

JP07049834

Publication Title:

No title available

Abstract:

Abstract not available for JP07049834

Data supplied from the esp@cenet database - Worldwide

Courtesy of <http://v3.espacenet.com>

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-49834

(43) 公開日 平成7年(1995)2月21日

(51) Int.Cl.⁶

G 0 6 F 15/00

識別記号

3 1 0 A 7459-5L

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数 1 O L (全 15 頁)

(21) 出願番号 特願平6-48803

(22) 出願日 平成6年(1994)3月18日

(31) 優先権主張番号 0 8 / 0 3 5 , 0 9 2

(32) 優先日 1993年3月19日

(33) 優先権主張国 米国 (U S)

(71) 出願人 592089054

エイ・ティ・アンド・ティ グローバル
インフォメーション ソリューションズ イン
ターナショナル インコーポレイテッド
アメリカ合衆国 45479 オハイオ、デイトン
サウス パターソン プールバード
1700

(72) 発明者 アリソン エー. カーレトン
アメリカ合衆国、イリノイ州 60622 シカゴ、リービット 1425

(74) 代理人 弁理士 西山 善章

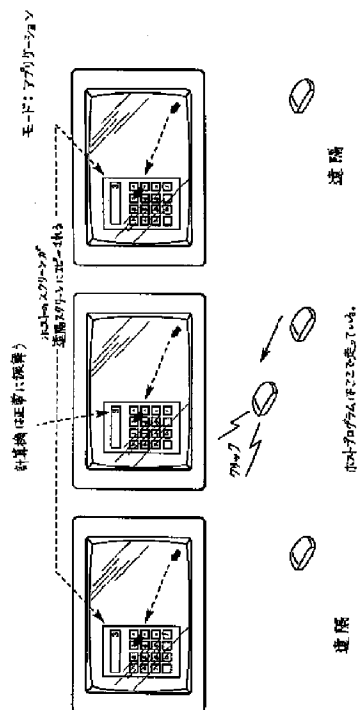
最終頁に続く

(54) 【発明の名称】 遠隔協同システム

(57) 【要約】

【目的】 会議を開くための複数コンピューターの使用法を与える。

【構成】 アプリケーションプログラムを単一のコンピューター上で走らせながら遠隔の参加者がそのプログラムに命令を発することができる。本発明は動作中のプログラムのディスプレイ表示を遠隔コンピューターのディスプレイ上に複製するので、遠隔の参加者はプログラムのオペレーションを観察することができる。参加者は誰でも自分のコンピューターディスプレイ上に添え書きをすることができる。本発明はその添え書きを他の参加者のディスプレイに複製する。



【特許請求の範囲】

【請求項1】 相互接続された一群のコンピューターにおいてその一つが他のコンピューターに格納されているプログラムを走らせるオペレーションの方法であって、

a) 各コンピューターを (1) 観察者、(2) 添え書き者、または (3) 制御者に指定するステップと、

b) いかなる観察者、添え書き者、または制御者のなすオペレーションに対していずれの観察者が影響も与えることをまったく許さないステップと、

c) 添え書き者達に自分のディスプレイに添え書きを許すと共に、観察者、添え書き者および制御者達のディスプレイに該添え書きの複製を作成するステップと、

d) 共有プログラムのオペレーションに対していずれの添え書き者も影響を与えることを許さないステップと、

e) 制御者に

i) 共有プログラムのオペレーションに影響を与えること、

ii) 添え書き者として行動すること

を許すステップとを含むことを特徴とするオペレーションの方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、複数のユーザーが、単一のコンピュータープログラムを遠隔操作することができるようにさせるシステムに関する。本発明は、すべてのコンピュータに分配される共通の視覚的イメージを発生する。ユーザーは共通のディスプレイ（表示）上に添え書きを作成することができる。ユーザーが添え書きを作成すると、本発明はその添え書き（annotation）をすべてのディスプレイ上に複製する。添え書きはユーザーによって個人的に秘密に保持することもできる。

【0002】

【従来の技術】 近代的電話システムは、異なる場所に居る複数のパーティーが会議を行なうことを可能にさせる。しかしながら電話会議は参加者すべてが会議室の共通のテーブルで面会する会議で得られるすべての便宜を与えることはできない。

【0003】 例えば会議室では参加者は話題に関わる図面あるいは製品等を見ることができる。その様な観察は電話会議では不可能である。

【0004】 本発明は、参加者が遠隔地に居ながらにして物理的に会議に参加しているときの便宜の多くを得られるようにするシステムを与える。

【0005】

【発明が解決しようとする課題】 本発明の課題は改良された電子会議システムを与えることである。

【0006】 本発明の別の課題は、ユーザーが遠隔的にコンピュータープログラムを作動させることができるようにするシステムを与えることである。

【0007】 本発明の更に別の課題は、複数のコンピュ

ータの一つに存在する単一のプログラムをそれらのコンピューターが作動させることができるようにするシステムを与えることである。

【0008】 本発明の更に別の課題は、複数のコンピューターのユーザーが共通の表示を見ることができるとともに、それに添え書きすることができるシステムを与えることである。

【0009】

【課題を解決するための手段】 本発明はその一形態として多重コンピュータを相互に接続する。アプリケーションプログラムは単一のコンピュータが走らせる。そのプログラムにより作成される表示は他のすべてのコンピュータのディスプレイ上に複製される。各コンピュータのユーザーは、使用している自分のディスプレイ上に添え書きすることができる。また本発明はそれらの添え書きを他のすべてのディスプレイ上に複製することができる。本発明は一コンピュータのユーザーが他人のコンピュータ上にロードされているプログラムを走らせることができる。

【0010】

【実施例】

概観

図1は電話線で接続された三台のコンピュータを示す。各コンピュータは、メッセージ駆動されるマルチタスキンググラフィカルユーザーインターフェイス（Graphical User Interface, GUI）を走らせる。その一例はワシントン州レッドモンド市のマイクロソフトコーポレーションから市販されているウィンドウズという名のものがある。そのような GUI はまたオペレーティング動作環境（operating environments）とも呼ばれる。

【0011】 GUI のユーザーはウィンドウズを使ったプログラムを介して相互作用を与えることができる。本発明は遠方にあるコンピュータのディスプレイ全体ではなくて、選択したいくつかのウィンドウを複製する。この選択的な複製によってユーザーは自分のディスプレイ（共用されていない自分のエリア）上に個人的エリアを維持することが可能となる。各コンピュータはまた本発明者が開発したソフトウェアを走らせることができる。さらに、一つのコンピュータ（ホスト）がアプリケーションプログラムを走らせる。（GUI のマルチタスキング能力があるため、ホストが 両方のプログラムを走らせることが可能である）。

【0012】 本発明は4つの基本モードのオペレーションを有する。

【0013】 1. アプリケーションモード（Application Mode）

図1に示す三台のコンピュータのいずれのユーザーもアプリケーションプログラムに命令を出すことができる。例えば、アプリケーションプログラムが手持ち型計算機をシミュレートするものであると仮定しよう。その初め

の状態が図2に示してある。この場合、各コンピュータディスプレイが計算機を図示する。次の事象が起こると仮定する：

・ホストのユーザーが（計算機プログラムの設計に応じてキーボードの入力あるいはマウスの入力により）計算機のボタン「3」を押す。これに応答して、各計算機がその表示領域に「3」を表示する。

【0014】・遠隔使用者の一人が「+」を押す。他の遠隔ユーザーが「6」を押す。

【0015】・ホストユーザーが「=」を押す。

【0016】この時点ですべての計算機が3と6の和である「9」を表示する。ユーザー達は集団でこの計算機プログラムのオペレーションを行い、各自のディスプレイがその結果を表示する。

【0017】この計算機プログラムは、合法的なシーケンスが受信される限り、いずれのユーザーがボタンを押したかは考慮せず、またユーザーの誰かがボタンを押さなかったかも考慮しない。（ただしユーザーたちは協力的であり、いずれのユーザーも計算機のオペレーションを懈怠しないと仮定する）。

【0018】2. 添え書きモード (Annotation Mode)

いずれのユーザーも「ペイント」プログラムに見られるものと同様の作図ツールを使ってユーザー自身のローカルディスプレイ上に描画することができる。ユーザーは、ボックス、円、円弧、楕円等を作図することができる。ユーザーはまたディスプレイ上の項目を消去することができる。

【0019】本発明はユーザー達の添え書きを他のすべてのディスプレイ上に複製することができ、したがってすべてのユーザーが類似のディスプレイを見ることができ、しかし次のような因子があるので、それらの表示は異なることがあり得る。

【0020】(A) ディスプレーモニターが異なれば、色の表現能力等の性質が異なる。

【0021】(B) ディスプレーのプロトコル (IGA, BGA など) が異なればグラフィックイメージの表現が異なり、表示能力が異なる。

【0022】(C) GUI が異なり、あるいは同一の GUI でもバージョンの異なる場合は表示方式が異なることがある。

【0023】(D) ユーザーの何人かは計算機が表示されるウィンドウの大きさを変更することがあり、その結果縮尺に差異が生じていることがある。

【0024】これらの差異は表示されるイメージの外観に差を生じさせる。しかしすべてのディスプレイの基本的な内容は同一なはずである。寸法の差異を許容するために、本発明は適当な縮尺で作図をすることができる。

【0025】3. ローカル添え書きモード (Local Annotation Mode)

ユーザーはローカルディスプレイに添え書きすることが

できるが、それらの添え書きは秘密に保たれるので、他のユーザーはその添え書きを見ることができない。

【0026】4. 観察モード (View Mode)

いずれのユーザーも添え書きすることができず、命令を発行することもできない。しかし添え書きに類似した行動を取ることができる。ユーザーは自分たちのカーソルを移動させることができ、他のユーザーはその動きを見ることができるので、遠隔的指摘（カーソルによる表示物の位置の指摘）ができる。観察モードは、例えば添え書きモードの実行中にある特定ユーザーのモードを観察モードに指定する実施例などで有用である。この実施例ではすべてのユーザーが添え書きすることができるが「観察」ユーザーは見ることができるだけであり、添え書きすることができない。

【0027】各モードの説明

図3-図14は例として計算機プログラムの使用を通して色々なモードを図示する。

【0028】図3

ホストがアプリケーションプログラムを走らせる。

【0029】モードは「添え書き」である。

【0030】ユーザーの入力はホストコンピュータで行なわれる。

【0031】ユーザーが計算機のオペレーションを試みる。

【0032】ホストコンピュータのユーザーが計算機を使って二つの数の加算を試みると仮定する。その試みの対象となる最初の数について考える。

【0033】ホストユーザーはホストのカーソルを計算機のキー上に移動させ、マウスをクリックすることによりキー入力を試みる。しかしマウスのクリックはアプリケーションプログラムに届かない。なぜならば本発明がそれを阻止するからである。アプリケーションプログラムは応答しない。なぜならばプログラムがマウスのクリックを受理しないからである。

【0034】すなわち、さらに詳細に説明すると、GUIはこのマウスの移動を検出し、発生すべき「マウスメッセージ」を発生させる。GUIはこのマウスメッセージをキュー (queue) におく。マウスメッセージはこのキューの中で処理を待機する。図15に示す「入力ルーター (INPUT ROUTER)」がこれらのメッセージを読む。

「添え書きモード」が現在実行中であるので、入力ルーターはこのメッセージを添え書きブロックに送る。アプリケーションはこのメッセージを受理せず、従って応答しない。このマウスクリックは無視される。

【0035】「添え書き」の応答

「添え書き」はマウスのメッセージに対して2通り（あるいはそれ以上）の応答をするように構成することができる。その一つの構成では、添え書きはマウスが初めに添え書きツールを選択することを要求する。選択がまったく行われないと、「添え書き」はマウスメッセージを

無視する。

【0036】選択は通常「ペイント」プログラムで行なわれるようにツールイメージ上でマウスをクリックすることにより行なう。「添え書き」はこのツールの選択を認識し、次いでそれ以降のマウスのクリックは選択ツールで描画するためのデータとして扱う。例えば長方形ツールが選択されたときは、次の二つのマウスのクリックが長方形の対角線上の隅を確定する。(後で議論する図4が長方形の描画を例示する。)第二の構成では、添え書きモードにあるときはペンのようなデフォルトツールが自動的に選択される。この構成ではユーザーが(クリックすることにより)計算機のボタンを押す試みを行うと、そのペンを使ったユーザーの線の描画が(意図せずに)始まる。これを認識したユーザーは公知方法のいずれかを使って線の描画を修正させることができる。

【0037】それゆえ添え書きモードでは本発明は(a)デフォルトの「添え書き」を開始することによってマウスの入力に応答する、あるいは(b)「添え書き」ツールが選択されないためにマウスの入力を無視する、のいずれかの応答をする。ユーザーからのキーボード入力も同様に扱われる。もちろん「添え書き」が他の応答をするように設計することもできる。

【0038】カーソルの追跡

各ディスプレイは関連のマウスで位置決定できるカーソルを示す。本発明はすべてのディスプレイ上に各カーソルを複製することができる。この様にして図3に示すように3つマウスを使うとき、3つのカーソルがある。(ただし簡単のため、ただ一つのみ図示してある。)

その結果、一人のユーザーがマウスを動かすとそれに対応するカーソルがすべてのディスプレイ上で運動する。

【0039】一般的に3つのカーソルは識別が可能である:各カーソルは例えば色・形状・ラベルの導入等によりその所有者を同定する。

【0040】図4

ホストがアプリケーションプログラムを走らせる。

【0041】モードは添え書きモードである。

【0042】ユーザーの入力はホストコンピュータで行なう。

【0043】ユーザーは計算機の上にボックスの描画を試みる。

【0044】この状況は図3に示すものと極めて類似している。ただし今の場合ユーザーが図3においてはボタンを押すことを意図している代わりに添え書きを描画しようとしている点が異なる。

【0045】ホストコンピュータのユーザーが計算機の上にボックスを描画すると仮定する。(このボックスは強調のため過度に大きく図示されている。ボックスは計算機自身を越えて伸びないことが望ましい。)本発明は遠隔のコンピュータ上にそのボックスを複製する。

(このボックスは図示していない添え書きツールを使っ

て描かれる)。

【0046】図15で見て、入力ルーターは論理の流れを「添え書き」宛てに送る。「添え書き」はそのボックスを描くための適当なGDI機能呼び出す。また「添え書き」は「添え書きメッセージ」を接続APIに送る。この接続APIが遠隔者に添え書きメッセージを届ける。

【0047】図15Aに示す「添え書き」はこの添え書きメッセージを受信する。この「添え書き」ブロックは各遠隔コンピュータにおいて実行される論理を表す。この「添え書き」はブロックGDIを経由して適当なGDI機能呼び出す。

【0048】「GDI」とはグラフィカル デバイス インターフェイスの略号である。「GDI機能」はGDI.EXEと呼ばれる大きなプログラムに含まれている。GDI機能は、呼び出されると、その後でユーザーが入力する入力に基づいて円、ボックスあるいはテキストのような特定のグラフィックイメージを描く。ほかのGDI機能はペン幅の選択などの他のタスクを行なう。

【0049】GDI.EXEは市販されている製品である。GDI.EXEに関する技術的な詳細はマイクロソフト社から入手可能な「ウィンドウズ ソフトウェア開発キット」およびチャールズ ベトゾルド著の「ウィンドウ図3.1のプログラミング」(ワシントン州レッドモンド市:マイクロソフト プレス、ISBN 1-55615-395-3、1992年発行)に記載されている。

【0050】図5

ホストがアプリケーションプログラムを走らせる。

【0051】モードは「アプリケーション」である。

【0052】ユーザー入力はホストコンピュータで行なわれる。

【0053】ユーザーは計算機の使用を試みる。

【0054】ホストユーザーは計算機のキー「3」の上にカーソルを移動させマウスをクリックする。GUIはマウスメッセージを発生し、それをキューにおく。本発明はマウスメッセージを呼んで、そのメッセージをアプリケーションプログラム(すなわち計算機プログラム)に送る。このプログラムは、(1)キー「3」が押されたこと、及び(2)GDIのコールを使って計算機のディスプレイに数字「3」を描画する、という応答をする。

【0055】このアプリケーションプログラムはまた、そのプログラムの内部的オペレーションを行うため、ユーザーが「3」を入力したという事実を記録する。

【0056】本発明は、計算機の中に「3」を描画している際、および押されている「3」のボタンを表示する際に、アプリケーションプログラムが出すGDIコールを中断する。他のコンピュータは同一のGDI機能を実行することによりホストディスプレイを複製する。このGDI中断に関する一層詳細な説明は「一般的な考

察」と題する後の節で述べる。

【0057】この様にして、ホストのユーザーが計算機を動作させることを全てのユーザーが同時に見ることができる。（きわめて短い遅延が関与するので、このホストの動作は厳密に同時的ではない。とは言え、ホスト及び遠隔のコンピューターが並んで動作したとしても人は多分その遅延に気付くことができないであろう。）図15においてマウスメッセージがアプリケーションプログラムに指向されるべきであるということを入力ルーターが認識し、論理の流れをアプリケーション（すなわち計算機プログラム）に指向する。アプリケーションは、

（1）押された「3」のキーを描画し、（2）適当なGDI機能呼び出す、ことによって計算機のディスプレイ内に数字「3」を書き込む。

【0058】しかし本発明はそれらが実行される前に、アプリケーションプログラムのGDI呼び出しを図15に示すGDIキャプチャー経由で捕捉（キャプチャー）する。本発明は捕捉した呼び出し（コール）に付いてふたつのことを行なう。その一つとして、本発明は「接続API」ブロックを経由してこれらの呼び出しを他のコンピューターに通告する。このアクションは図15Aに示す「捕捉されたGDIディスプレイ」ブロックに届く。そうするとブロックGDIによって示すようにこのブロックが各遠隔者に同一のGDI機能を実行させる。

【0059】その二として、図15に示すブロックGDI経由で本発明は、ホストで実行すべきアプリケーションプログラムが呼び出すGDI機能をホストで実行することを許す。

【0060】それゆえ、本発明はアプリケーションプログラムが行なったGDI機能の呼び出し（コール）を捕捉する。本発明は捕捉したコールを遠隔コンピューターに通告する結果、遠隔コンピューターはそれらのコールを複製することができる。本発明は、その捕捉したコールが意図されたようにホスト上で実行されることを許す。

【0061】図6

ホストがアプリケーションプログラムを走らせる。

【0062】モードは「ローカルな添え書き」である。

【0063】ユーザー入力力はホストコンピューターで行なわれる。

【0064】ユーザーが計算機に演算を行なわせることを試みる。

【0065】添え書きモードにおいて、ユーザーに与えられているデフォルト添え書きツールがまったく無いと仮定しよう。この仮定の下で、ユーザーがカーソルを計算機のボタンまで移動させてボタンを「押す」と、図15の「入力ルーター」がマウスメッセージを「添え書き」ブロックまで通過させる。このマウスクリックは有効な添え書き入力シーケンスではないではない（ツールが何も選択されていない）ので、何の「添え書き」も描

かない。

【0066】更に、遠隔コンピューターは、ホストコンピューターのマウスに対応するカーソルの移動を示さない。なぜならば前述したように、「ローカルな添え書き」の実行中は図15のライン5が添え書きメッセージを他のコンピューターに送らないからである。

【0067】更にまた、ホストディスプレイ上で計算機のボタンを押しても、それに応答して押されたボタンを描き直すことはしない。なぜならば、アプリケーションがマウスメッセージを受信しなかったからである。アプリケーションが責任をもつのは押された計算機ボタンの描画である。

【0068】添え書きのデフォルトがローカル添え書きモードをとっているユーザーに指定されているときは、そのユーザーのマウスクリックによってそのツールによる描画が開始される。ユーザーが誤りに気づいたとき、ユーザーは公知の方法で描画を終了するだろう。

【0069】図7

ホストがアプリケーションプログラムを走らせる。

【0070】モードは「ローカルの添え書き」である。

【0071】ユーザー入力はホストコンピューターで行なわれる。

【0072】ユーザーは計算機に添え書きを試みる。

【0073】これらの条件のもとで、図15のA部に示す入力ルーターはボックスの描画などの有効な添え書きを認識する。入力ルーターはその論理の流れを「添え書き」ブロックに指向し、「添え書き」ブロックが図7に示すボックスの描画等の添え書きのための適当なGDI機能呼び出す。

【0074】しかし、添え書きがローカルであるので、図7に示すように遠隔コンピューター上には何のボックスも描かれない。図15のデータ路5に沿って何のデータも送られない。

【0075】図8

ホストがアプリケーションプログラムを走らせる。

【0076】モードは「観察」である。

【0077】ユーザー入力はホストコンピューターで行う。

【0078】ユーザーが計算機の実行を試みる。

【0079】図8が示すようにマウスのクリックは無視され、遠隔コンピューターには何にも起きない。

【0080】図15において入力ルーターはこのマウスメッセージを読むが、アプリケーションに届かないようにそれを阻止する。なぜなら現在のモードが「観察モード（View Mode）」だからである。

【0081】図9

ホストがアプリケーションプログラムを走らせる。

【0082】モードは「添え書き」である。

【0083】ユーザー入力は遠隔コンピューターで行な

う。

【0084】ユーザーが計算機の実操作を試みる。

【0085】ユーザーがマウスカーソルを計算機のボタン上に移動させ、マウスをクリックすると仮定する。このクリックは無視される。他のコンピュータ（ホスト及び他の遠隔コンピュータ）はユーザーのカーソルの運動を示すが、それ以外は何も示さない。その理由は、ツールが何も選択されていないからである。

【0086】図15Aにおいて、入力ルーターはこのマウスメッセージがアプリケーションに到達することを阻止する。論理演算は「添え書き」に当てられ、GDIブロックを経由して「添え書き」がユーザーの遠隔ディスプレイ上にカーソルを描画する。「添え書き」もまた接続APIにデータを送り、この接続APIが論理演算を図15の「添え書き」に当てる。この「添え書き」は、他の二つのコンピュータ（すなわちホスト及び当該他の遠隔コンピュータ）上に存在する添え書き論理内容を表す。これらの「添え書き」ブロックはそれぞれのユーザーのカーソルに対応するカーソルを、図15に示すGDIブロックを経由してそれぞれの場所で描画する。図15のGDIはGDI機能コールの内容を表わす。

【0087】ホストがボックス描画ツールのような一つのツールを使うことができる一方、遠隔コンピュータは円を描くツールのような別のツールを使用することができる。

【0088】図10

ホストがアプリケーションプログラムを走らせる。

【0089】モードは「添え書き」である。

【0090】ユーザー入力は遠隔コンピュータで行なう。

【0091】ユーザーが計算機への添え書きを試みる。

【0092】添え書きはボックスであると仮定する。全てのディスプレイ上にボックスが描かれる。図15Aにおいて、ユーザーの遠隔コンピュータにある入力ルーターがマウスメッセージを「添え書き」ブロックに送る。「添え書き」は二つのことを行なう。その一つは「添え書き」が添え書きであるボックスの描画を行なうための適当なGDIを呼び出す。

【0093】その二は、「添え書き」が接続APIに添え書きメッセージを送り、接続APIが他方のコンピュータに添え書きメッセージを送る。ただしこれらの一方はホストコンピュータであり、他方は遠隔コンピュータである。ホストの論理プロセスが図15の「添え書き」に到達し、前記他方の遠隔コンピュータの論理プロセスが図15Aにある「添え書き」に到達する。

【0094】これらの「添え書き」ブロックは両者とも、適当なGDI機能呼び出してこれにユーザー所望の添え書きを描かせる。しかしホストの論理経路5はこの時点で取り上げられない。なぜなら、ホストの添え書

きを他のコンピュータに複製する必要がないからである。

【0095】図11

ホストがアプリケーションプログラムを走らせる。

【0096】モードはアプリケーションである。

【0097】ユーザー入力は遠隔コンピュータで行う。

【0098】ユーザーが計算機の実操作を試みる。

【0099】読者はこの計算機プログラムがホストにのみロードされていることを思い起こされたい。ただし遠隔コンピュータのユーザーがそれを作動させたいと希望している。

【0100】図15Aに示す遠隔ユーザーの入力ルーターは、マウスメッセージを接続APIに送る。ホストはこれらのメッセージを受信し、これらのメッセージは図15に示すホストの入力ルーターに届けられる。ホストの入力ルーターはこれらメッセージをブロックアプリケーション（すなわち計算機プログラムであるアプリケーションプログラムに）送る。このプログラムは二つの重要なことを行なう。

【0101】遠隔マウスがそれらメッセージ出したにも関わらず、計算機プログラムはそれらのメッセージをあたかもホストのマウスによって発行されたかのように扱う。計算機プログラムはユーザーがデータ（すなわち「3」）を入力したことを知ると、通常の方法で応答する。その応答は次のことを含んでいる：（1）押された計算機ボタン「3」を示すこと。（2）計算機のディスプレイ内に数字「3」を書き込むこと。及び（3）それ自身の内部的計算を行なうこと。

【0102】しかし、計算機プログラムが前節の（1）（2）を実行することができるようになるまえに、本発明は最初に計算機プログラムが呼び出すGDI機能捕捉する。この捕捉は図15のGDI捕捉ブロックに例示されている。この捕捉の際、本発明は二つのことを行なう。その一つは、（他のコンピュータが使用できるようにするため）本発明がこれらのGDI機能を接続APIに送る。ユーザーの遠隔コンピュータにおいては、図15Aに示す接続APIがこのGDI機能を「捕捉されたGDIディスプレイ」に送る。このディスプレイはホストディスプレイの複製を作成する。その二として本発明は（図15に示すブロックGDIを経由して）ホストで実行されるべきGDI機能をホストで実行させる。それゆえ、事象の一般的シーケンスは次の通りである：

・遠隔ユーザーが計算機ボタンを押す試みをする。

【0103】・遠隔コンピュータ上で作動している本発明は、この試みを検出し、ホスト上で走っている計算機プログラムにデータを送る。このデータはメッセージの形式を取る。計算機のプログラムはこのメッセージがホストのマウスからきたものと「考える」。

【0104】・計算機プログラムが通常通りに働き、GDIコールを経由してホストディスプレイ上にイメージを描く。

【0105】・本発明がGDIコールを捕捉し、それらを遠隔コンピュータに通報する。

【0106】・遠隔コンピュータはホストのウィンドウを複製する。遠隔ユーザーはこのようにして、ホスト上で走っている計算機プログラムを遠隔的に作動させることができる。

【0107】別の言い方で要約すると、本発明は遠隔コンピュータのマウスメッセージに基づいてマウスメッセージをホストに発生する。(ホスト上で走っている)計算機プログラムが、それらのマウスメッセージがあたかもホストで発生されたかのようにマウスメッセージに応答する。本発明は計算機プログラムが行ったGDIコールを中断し、遠隔コンピュータにおける同一のGDIコールを実行することにより、ホストディスプレイを遠隔コンピュータに複製する。

【0108】図12

ホストがアプリケーションプログラムを走らせる。

【0109】モードは「ローカルな添え書き」である。

【0110】ユーザー入力は遠隔コンピュータで行なう。

【0111】ユーザーが計算機の実行を試みる。

【0112】当該ユーザーのマウスクリックは無視される。ツールの選択に失敗しているため、このマウスの運動に応答して他のディスプレイ上に何も現れない。

【0113】図13

ホストがアプリケーションプログラムを走らせる。

【0114】モードは「ローカルな添え書き」である。

【0115】ユーザー入力は遠隔コンピュータで行なわれる。

【0116】ユーザーが計算機への添え書きを試みる。

【0117】上に記したように、添え書きはユーザーのディスプレイ上に描かれる。他のディスプレイ上には何の添え書きも起きない。

【0118】図14

ホストがアプリケーションプログラムを走らせる。

【0119】モードは「観察」である。

【0120】ユーザー入力は遠隔コンピュータで行なう。

【0121】ユーザーが計算機の実行を試みる。

【0122】ユーザーのディスプレイ上でマウスのカーソルが移動するが、マウスクリックは無視される。また他の二つのディスプレイはユーザーのマウスカーソルの運動を示さない。

【0123】一般的考察

1. 別異のプログラムはディスプレイ全体のうちの異な

る部分で描画を行う。ディスプレイはGDI機能を使って描かれる。しかし一つのディスプレイでも異なる部分はそれぞれ別のプログラムによって描かれる。

【0124】これらすべての描画オペレーションがGDI機能を使用して行なわれるにも拘わらず、GDI機能はコンピュータ間の排他的ディスプレイ複製通信媒体ではない。

【0125】添え書きは一つの形態のコンピュータ間データ転送を含む。

【0126】アプリケーションプログラムによる描画は他の形態のものを含む。

【0127】例えば、ユーザーが添え書きを行なうとき、ユーザーのマウスメッセージは図15に示す経路5を経由して、他のコンピュータにおいて「メッセージ」として複製される。これらの複製メッセージは次いで、(他のコンピュータにおける)それぞれの「添え書き」ブロックがその添え書きを描くための適当なGDIコールを発行させる。すなわちGDIコールは、その添え書きを行なっているユーザーからは直接に他のコンピュータに送られない。

【0128】それとは対照的に、アプリケーションプログラムがグラフィックイメージをディスプレイ上に描かせている場合、本発明はそのGDIコールを(図15に示す「GDI捕捉」を経由して)中断し、そのGDIコールを他のコンピュータ上に複製させる。

【0129】相違の理由

ふたつの異なる手順(マウスメッセージの複製及びGDIコールの複製)を行なう主な理由は、添え書きがディスプレイ情報とは異なった位置に格納されるためである。

【0130】すなわち図2の計算機に戻って、アプリケーションプログラムは次のような一般的な方法で計算機のイメージを格納する。添え書きデータは本発明によって格納される。アプリケーションプログラムデータは(ホストにある)アプリケーションプログラムによって格納される。各キーのイメージはGDI機能がそのキーを引き出すことのできるデータとして格納される。このデータは位置、大きさ、色等の情報を含む。各キーは関連の数を含む。その数は位置・大きさ・フォントタイプなどの情報とともにテキストキャラクターとして格納することができる。

【0131】添え書きデータは異なった場所に格納されるが同じ一般的な方法で格納される。

【0132】添え書きあるいはアプリケーションプログラムのいずれかがビットマップを必要とするときは公知の方法でビットマップがGUIにより格納される。

【0133】本発明は、マスキングに関する公知の技術によって添え書きイメージをアプリケーションのイメージと組み合わせる。すなわち本発明は、遠隔コンピュータが受信したGDI機能をビットマップに再生する。

本発明は、受信した添え書き情報を、それとは異なるビットマップに再生する。これら二つのビットマップは相互にマスク掛けされる。

【0134】ユーザーがアプリケーションイメージを添え書き抜きで保存できるようにする等の目的で、添え書きデータはアプリケーションデータから分離して保存される。この代わりとして、ユーザーは添え書きデータを単独に、あるいは添え書き済み表示の形で、保存することができる。

【0135】別個保存するもう一つの理由は、添え書きデータを別個に保存することは添え書きデータを全く有しないディスプレイを描くのに便利であるからである。添え書きデータが計算機イメージデータと混和してしまったときは、添え書きデータを除去することは不可能でないまでも困難である。

【0136】もしもGDIコールを排他的に送信するとすると（すなわちメッセージの複製が全く行なわれないとすると）、添え書きデータを別個の格納するように構成するための余分の作業が必要となろう。

【0137】2. GDI の中断もしくは捕捉
GDI の中断は次のように理解することができる。

【0138】A. スタートアップの時点で、本発明は各GDI機能の最初の5バイトを、ある特定のプログラムすなわち Trap.GDI へのジャンプ指令 (JUMP) で置換する。

【0139】B. Trap.GDI は、所望のグラフィックイメージに対するパラメーターを獲得する（例えばボックスの場合はふたつの対角線の角の位置など）。そしてサブプログラムであるPkgDispCallを呼び出す。Trap.GDI はまた最初の5バイトをも置換する。

【0140】C. PkgDispCall は、Trap.GDI からパラメーターを受信してオブジェクト構造体 (object structure) を発生する。このオブジェクト構造体はほかのコンピューターがボックスを描くのに必要な全ての要素を含んだデータブロックである。

【0141】例えば、オブジェクト構造体はボックスの寸法及び位置に関する情報を含んでいる。更にGUIは、「コンテキスト (context)」内にイメージを描く。このコンテキストは、ペンの巾・色そのほかの特徴的なものを含んでいる。本発明は各コンピューターのコンテキストを追跡する。もしも描いたボックスのコンテキストが遠隔コンピューターのコンテキストと異なっていると、他のコンピューターが正しいコンテキストを発生するのに必要なデータをPkgDispCall が含める。

【0142】D. オブジェクト構造体は他のコンピューターに発送され、当該他のコンピューターが同一のGDI機能を実行する。

【0143】E. 本発明はオリジナルのGDI機能を実行する。

【0144】3. ディスプレイ全体が転送されることは

ない。ディスプレイはビット単位で置換されることはない。例えば、図2に示す計算機のイメージはビット単位でコンピューター間を転送することもできよう。もしも計算機が200×300ピクセルの空間を占めるならば、6万ピクセル（すなわち200×300のピクセル）に関する情報が送られなければならない。この代わりとして、図2に示す特定の計算機は18個の長方形および18個の長方形各々に対する一テキスト文字として扱われ、その結果全部で34個のオブジェクトとなる。各オブジェクトは、大きさ及び位置のようなパラメーターを必要とする。パラメーターの数は小さくて3乃至10である。10個のパラメーターがあると仮定すると、340個のデータを送らなければならない。もちろん、各データピース（データ片）の大きさは色々の因子に依存するが、各ピースには小さなバイト数を想定すれば足りる。

【0145】それゆえ、本発明はビット単位の複製では6万ピース必要なデータを、オブジェクト複製のための最大340ピースに低減する。もちろんオブジェクトによってはビットマップの形態を取ることがあり、ビットごとに送らなければならない。しかし一般的にビットマップは稀にしか現われない。更に、一般的にビットマップを送るときは一回だけ送られると予想される。

【0146】さらにオブジェクトデータは可能であれば圧縮される。すなわちコンピューター間の情報伝達は、可能であれば、圧縮したデータの形で行なわれる。圧縮は本技術分野で公知である。

【0147】4. データリンクの形態

コンピュータ間の通信はいくつかの形態を取り得る。市販されているネットワーク、ローカルエリア、及びワイドエリアを使うことができる。市販されているISDN電話回線は地方の電話会社から提供されており、これらを使うこともできる。モデム通信を使うこともできる。

【0148】5 先行技術のメッセージ検出

入力デバイスに応答してGUIで発生したメッセージを検出する市販のパッケージが利用できる。そのようなパッケージの一つは WINSIGHT であり、これはボーランド・インターナショナルから入手できる。しかしこれらのパッケージは遠隔コンピューターにメッセージ通報しないと思われる。

【0149】6 代わりとなる「GDI捕捉」

上記のグラフィックの捕捉に代わる方法は次の通りである。本システムが与えるGDIを、実際のシステムGDIを呼び出す前にGDIコールを処理する別個のプロシージャで置き換える。システムGDIの名前は二つのモジュール間で混同することを防止するために変更される。システムに提供されているモジュールを介して行なわれるGDIコールを捕捉するために、USR.EXE にも同じテクニックが使われる。

【0150】7. 一つ以上のコンピューターがアプリケ

ーションプログラムを走らせることができる。一つのプログラムに対してある特定のコンピューターがホストとして行動することができ、別のプログラムに対して遠隔コンピューターが行動することができる。例えば一つのコンピューターはワープロを走らせることができ、別のコンピューターは描画プログラムを走らせることができる。それぞれのプログラムに対しては各々がホストである。

【0151】各コンピューターに搭載された本発明のソフトウェアは同一もしくは実質上同一であるので、すべてのユーザーがワードプロセッサまたはCADプログラムを上記したように走らせることができる。

【0152】8. 「リアル」カーソルおよび「疑似」カーソル

二つのタイプの「カーソル」がある。各GUIはそれ自身の「リアル (real, 実際の)」カーソルを所有する。実際のカーソルはGUI機能によっては発生されず、GUIの中の独立の機能によって発生される。このカーソルは、マウスの運動にตอบสนองしてGUIが移動させるビットマップと見做すことができる。ローカルマウスによって制御されるそのリアルカーソルに加えて、本発明は各遠隔の参加者に対する「疑似」カーソルを発生する。疑似カーソルはGDI機能を使って発生される。時折りリアルカーソルはカーソルの移動とともに変化する。例えば、ツールバー上にあるときはカーソルは矢印の形状を取ることができ、クライアント領域上にあるときは手の形に変化することができる。しばしばこの変化はアプリケーションプログラムの制御のもとにおかれる。

【0153】それゆえ、(図11に示すように) 遠隔ユーザーがホストマシン上で走っているアプリケーションプログラムを制御するときは、GDIコールを使用せずにアプリケーションプログラムがホストマシン上のカーソルを変化させることができる。従って、図15及び15Aに示すGDIの捕捉は、変化したカーソルを遠隔ディスプレイ上で複製することに対して非効率である。

【0154】この問題に対処するため、本発明はリアルカーソル (例えば SetCursor コマンド等) を変化させる機能を監視する。本発明は遠隔コンピューター上でカーソル変化を複製する。

【0155】同一の SetCursor コマンドを実行することも一つの方法である。もう一つの代わりの方法は、適当なGDIコールのシーケンスを実行することによって、ホストカーソルが変化するとき遠隔カーソルを変化させ、またはビットマップを描かせることであろう。

【0156】9. ディスプレイ全体は複製されない。本発明はディスプレイのユーザーが同定するウィンドウを複製するだけである。すなわちユーザーはノートパッドのような作図片を会議の期間中秘密に保持することができる。GDIコールはタスクハンドル (task handle)

を使用する。共用するアイテムをタスクハンドルが参照しないならば、GDIコールは共用されない。

【0157】

【効果】本発明は相互接続された複数のコンピューターにおいて、ホストコンピューターのプログラムで制御された観察モード、添え書きモード、および制御モードにより、これら複数のコンピューターの各ユーザーが遠隔に居ながらにしてホストコンピューターのプログラムの制御に参加し、あるいは自己もしくは他のユーザーのコンピューターに添え書きを送り、他のユーザーのアクションを観察することを行うことができ、これらコンピューターユーザー間で会議類似の通信ができる。

【図面の簡単な説明】

【図1】 電話線で接続された三台のコンピュータを例示する図である。

【図2】 図の3ないし図14に詳細に説明する実施例の一例を示す。本例では計算機プログラムがいろいろのパーティーにより作動され、添え書きを付けられる。

【図3】 本発明が添え書きモードにあるときにホストユーザーの一人が計算機のオペレーションを試みる場合、本発明がどのように応答するかを示す図である。

【図4】 本発明が添え書きモードにあるときにホストユーザーの一人が計算機に添え書きを試みる場合、本発明がどのように応答するかを示す図である。

【図5】 本発明がアプリケーションモードのあるときにホストユーザーの一人が計算機のオペレーションを試みる場合、本発明がどのように応答するかを示す図である。

【図6】 本発明がローカル添え書きモードにあるときにホストユーザーの一人が計算機のオペレーションを試みる場合、本発明がどのように応答するかを示す図である。

【図7】 本発明がローカル添え書きモードにあるときにホストユーザーの一人が計算機に添え書きを試みる場合、本発明がどのように応答するかを示す図である。

【図8】 本発明がビューモードにあるときにホストユーザーにどのように応答するかを示す図である。

【図9】 本発明が添え書きモードにあるときに遠隔のユーザーが計算機のオペレーションを試みる場合、本発明がどのように応答するかを示す図である。

【図10】 本発明が添え書きモードにあるときに遠隔のユーザーが計算機に添え書きを試みる場合、本発明がどのように応答するかを示す図である。

【図11】 本発明がアプリケーションモードにあるときに遠隔のユーザーが計算機のオペレーションを試みる場合、本発明がどのように応答するかを示す図である。

【図12】 本発明がローカル添え書きモードにあるときに遠隔のユーザーが計算機のオペレーションを試みる場合、本発明がどのように応答するかを示す図である。

【図13】 本発明がローカル添え書きモードにあると

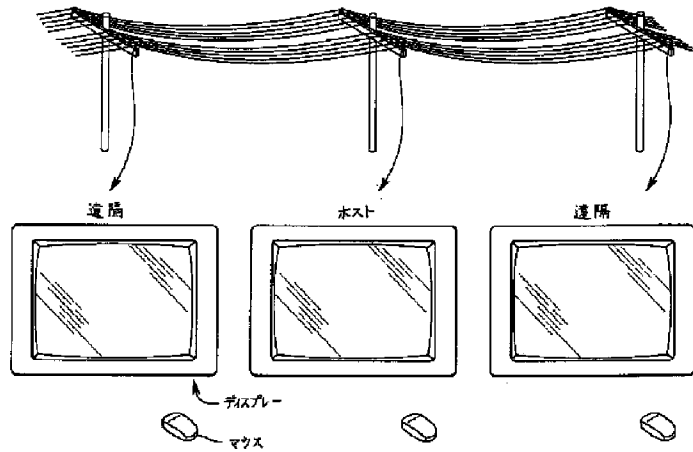
きに遠隔のユーザーが計算機に添え書きを試みる場合、
本発明がどのように応答するかを示す図である。

【図14】 本発明がビューモードにあるときに遠隔の

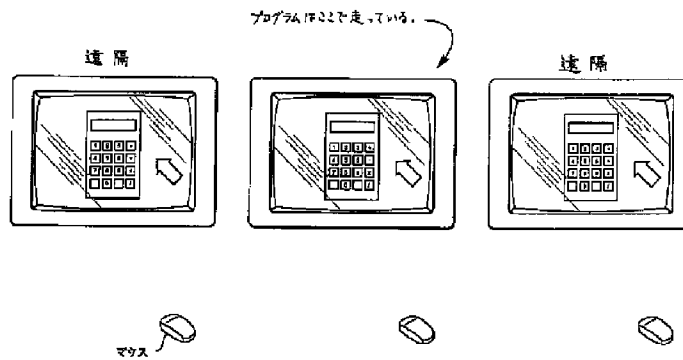
ユーザーにどのように応答するかを示す図である。

【図15】 本発明で使用する論理の流れを例示する図
である。

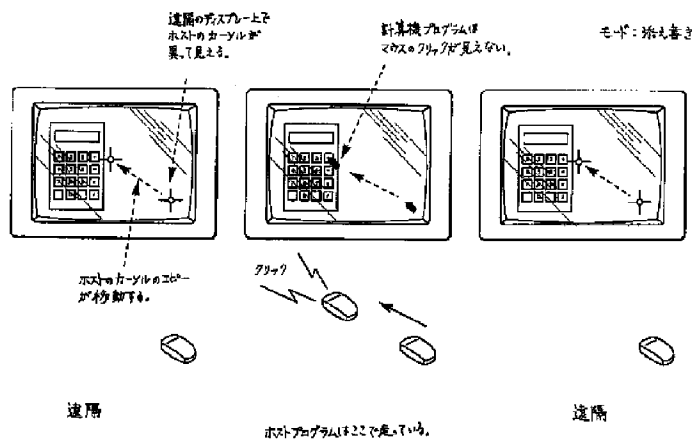
【図1】



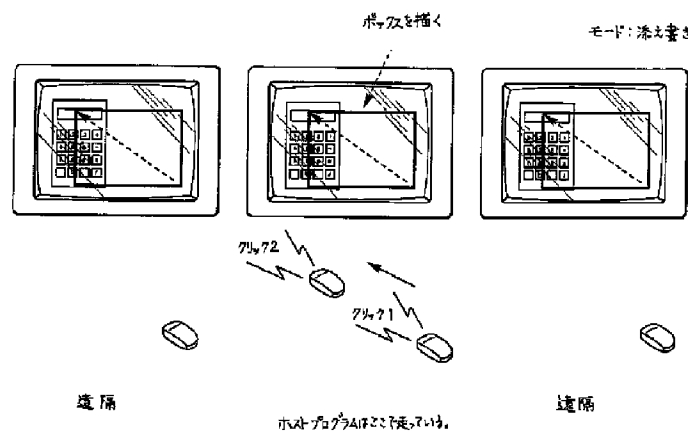
【図2】



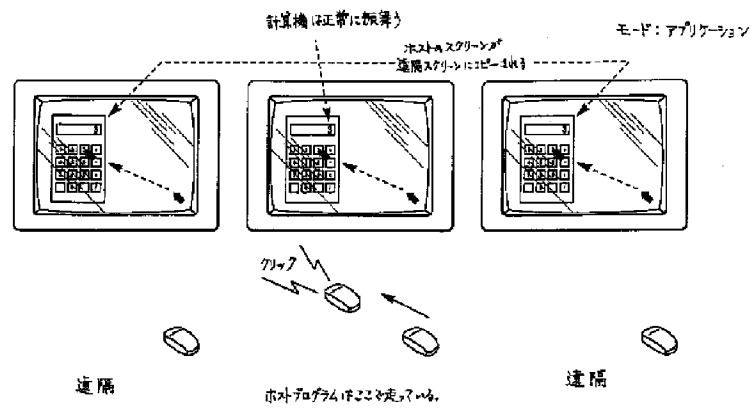
【図3】



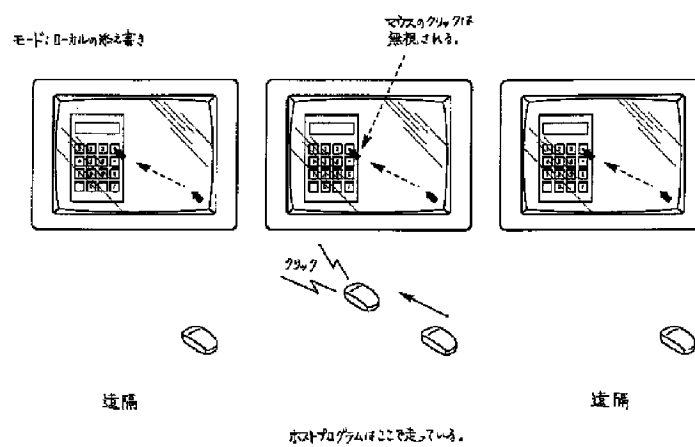
【図4】



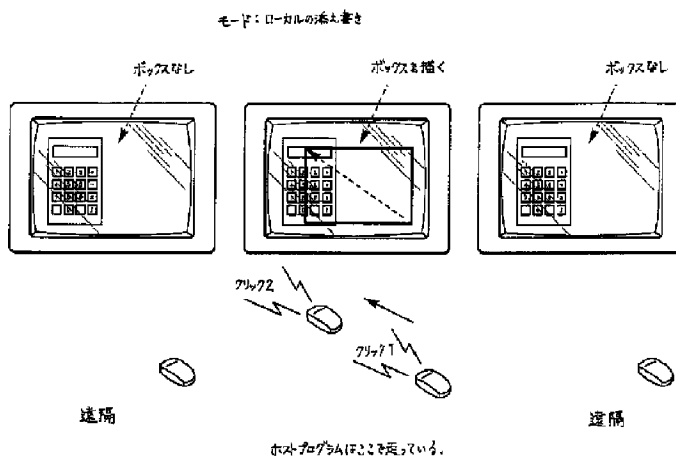
【図5】



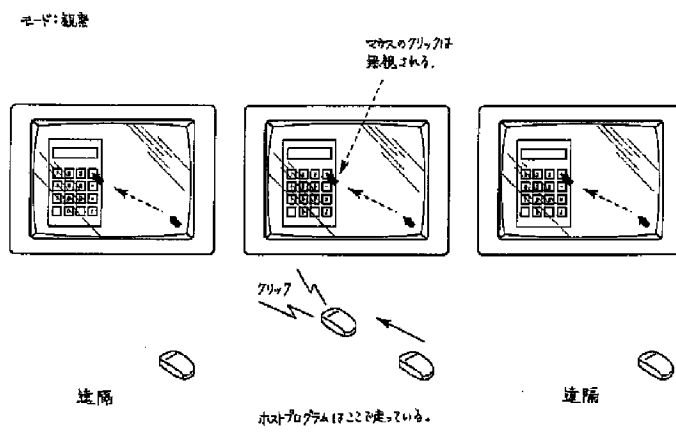
【図6】



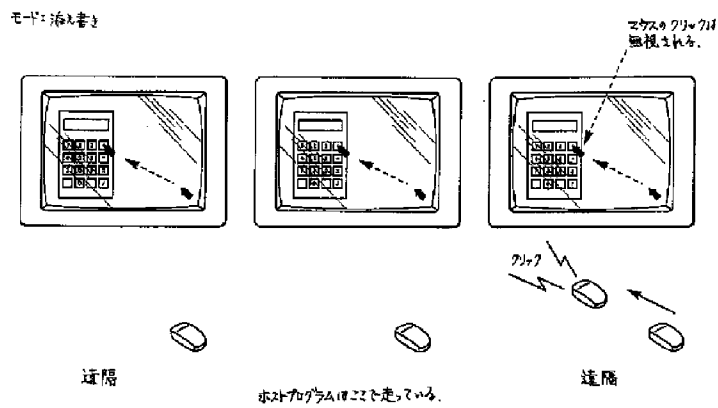
【図7】



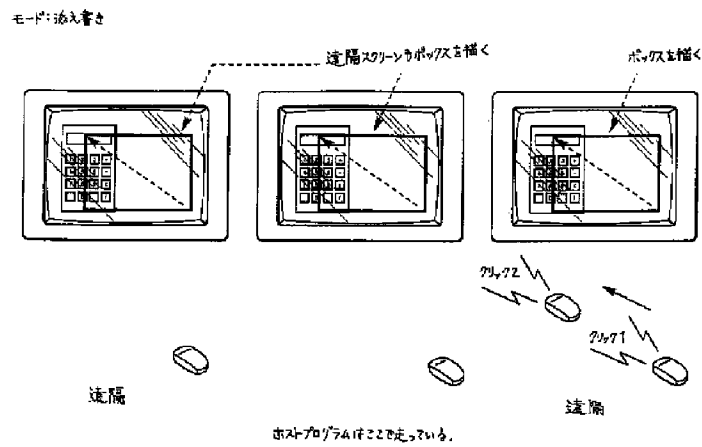
【図8】



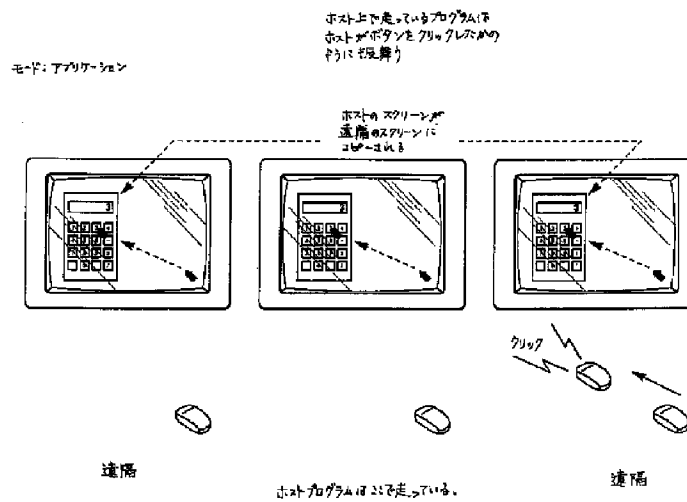
【図9】



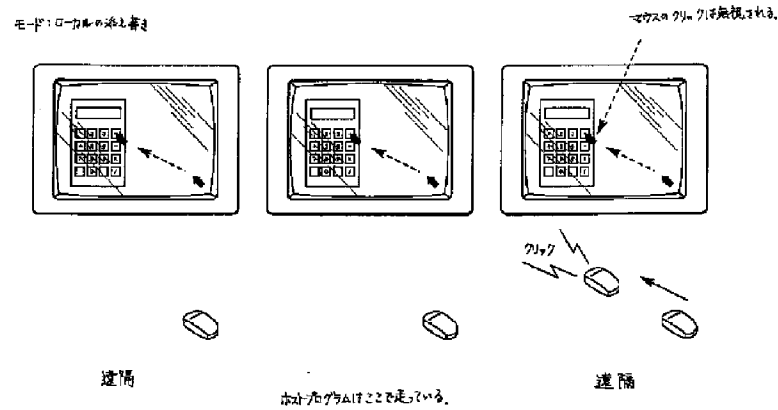
【図10】



【図11】

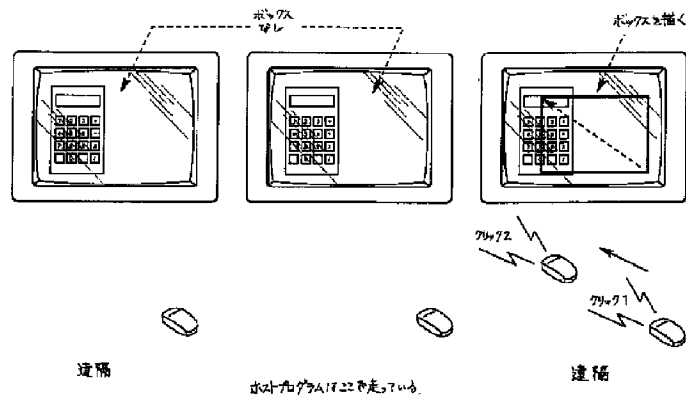


【図12】



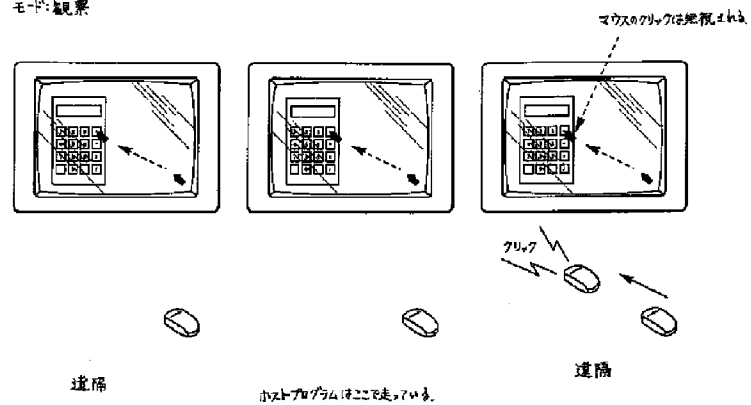
【图 13】

モード：ローカルの読み書き

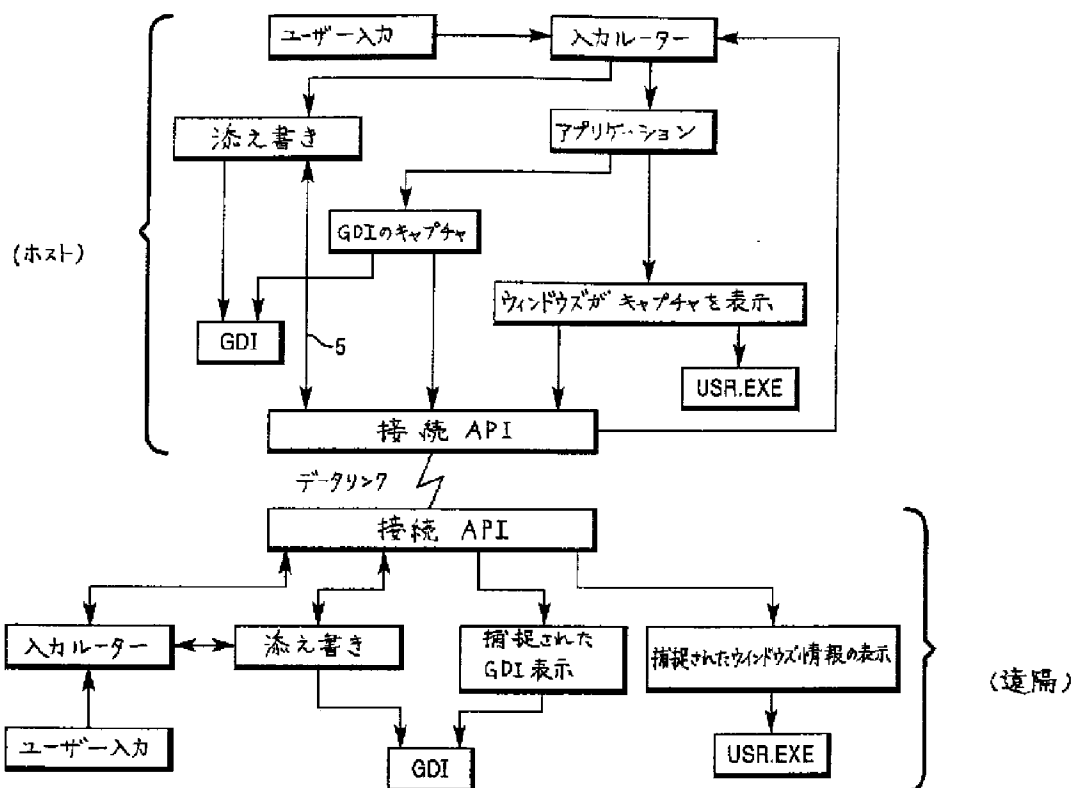


【图 1 4】

カード: 観衆



【図15】



フロントページの続き

(72)発明者 テリーサ エム. ポムア
アメリカ合衆国、イリノイ州 60559 ウ
ェストモント、サウス ウィルメット
126

(72)発明者 クリスタ エス. シュワルツ
アメリカ合衆国、イリノイ州 60510 パ
タビア、ジェニファー ドライブ 953
(72)発明者 キャサリン エム. フィツパトリック
アメリカ合衆国、イリノイ州 60510 パ
タビア、ジェニファー ドライブ 953

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 07-049834

(43)Date of publication of application : 21.02.1995

(51)Int.Cl.

G06F 15/00

(21)Application number : 06-048803

(71)Applicant : AT & T GLOBAL INF SOLUTIONS
INTERNATL INC

(22)Date of filing : 18.03.1994

(72)Inventor : CARLETON ALLISON A
POMMIER THERESA M
SCHWARTZ KRISTA S
FITZPATRICK CATHERINE M

(30)Priority

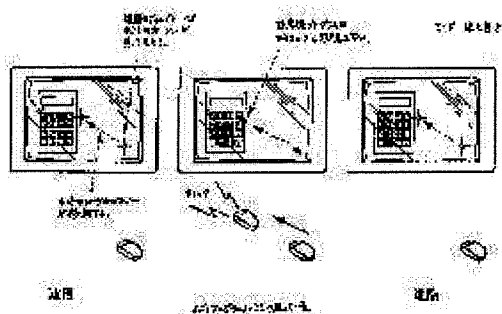
Priority number : 93 35092 Priority date : 19.03.1993 Priority country : US

(54) REMOTE COOPERATION SYSTEM

(57)Abstract:

PURPOSE: To apply a method for using plural computers for holding a conference.

CONSTITUTION: While running an application program on a simple computer, a participant at a distant place can issue an instruction to that program. Since the display indication of the program under operating is copied on the display of a remote computer, the participant at the distant place can observe the operation of the program. Every participant can additionally write information onto the display of that participant's own computer. That additionally written information is copied on the displays of the other participants.



TRANSLATION OF JAPANESE PATENT NO. JP07-049834

CLAIMS

[Claim(s)]

[Claim 1] a group by which interconnection was carried out -- one being the method of operation of running a program stored in the computer of other, and in a computer, a) A step which specifies each computer as (1) observer, (2) postscript person, or (3) control person, b) Allow [steps which do not allow an observer of a gap at all to receive operation which what kind of observer, a postscript person, or a control person makes, and to affect it, and c postscript persons] its own display a postscript, and. A step which creates a duplicate of this postscript on a display of observers, postscript persons, and control persons, d) A step which does not allow a postscript person of a gap to receive operation of a shared program and to affect it, either, e) A method of operation containing a step which allows a control person to affect operation of i shared program, and to act as an ii postscript person.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001] [Industrial Application] This invention relates to the system by which two or more users can make it enable it to operate a single computer program by remote control. This invention generates the common visual image distributed to all the computers. The user can create a postscript on a common display (display). If a user creates a postscript, this invention will reproduce the postscript (annotation) on all the displays. A postscript can also be individually held secretly by a user.

[0002] [Description of the Prior Art] A modernistic phone system enables two or more parties which are in a different place to hold a conference. However, the conference call can give no facilities obtained at the meeting in which all participants meet on the common table of a conference room.

[0003] For example, in a conference room, the participant can see a drawing or a product in connection with subject, etc. Such observation is impossible in a conference call.

[0004] This invention gives the system which enables it to obtain when [expedient / many of] the participant has participated in the meeting physically in the remote place.

[0005] [Problem(s) to be Solved by the Invention] The technical problem of this invention is giving the improved electronic meeting system.

[0006] Another technical problem of this invention is giving the system by which it enables it for a user to operate a computer program remotely.

[0007] Another technical problem of this invention is giving the system by which it enables it for those computers to operate the single program which exists in one of two or more of the computers.

[0008] Another technical problem of this invention is giving the system to which it can be added while the user of two or more computers can see a common display.

[0009] [Means for Solving the Problem] This invention connects a multiplex computer mutually as the one gestalt. A single computer runs an application program. A display created by the program is reproduced on a display of other computers of all the. The user of each computer can put a postscript on his display. [who is using it] This invention can reproduce those postscripts on other displays of all the. This invention can run a program by which a user of one computer is loaded on others' computer.

[0010] [Example] General-view drawing 1 shows three computers connected with the telephone wire. Each computer runs the multitasking graphical user interface (Graphical User Interface, GUI) by which a message drive is carried out. The example has a thing named the Windows marketed from Microsoft Corporation of Redmond, Washington. It is such. GUI is also called operating operating environment (operating environments) again.

[0011] the user of GUI gives an interaction via the program using Windows -- it can do. This invention reproduces not the whole display of the computer which exists far away but the window of shoes to choose. A user becomes possible [maintaining individual area on its own display (one's area which is not shared)] by this alternative duplicate. Each computer can run the software which this invention person developed again. One computer (host) runs an application program. (Since it has the multitasking ability of GUI, a host It is possible to run both programs).

[0012] This invention has the operation of four dominant modes.

[0013] 1. Application mode (Application Mode) Any user of three computers shown in drawing 1 can give a command to an application program. For example, an application program will assume that it is what simulates a stock type computer. The first state is shown in drawing 2. In this case, each computer display illustrates a computer. The user of : and the host who assumes that the following phenomenon happens pushes the button "3" of a computer (it responds to the design of a computer program and they are an input of a keyboard, or an input of a mouse). This is answered and each computer displays "3" on the viewing area.

[0014]- One of remote users pushes "+." Other remote users push "6."

[0015]- A host user pushes "=".

[0016] At this time, all the computers display "9" which is the sum of 3 and 6. Users perform operation of this computer program in a group, and each one of displays display that result.

[0017] This computer program does not take into consideration whether it did not take into consideration whether which user pushed the button, and someone of users pushed the button, as long as a lawful sequence is received. (However, it is assumed that users are cooperative and neither of the users does nonfeasance of the operation of a computer).

[0018] 2. Postscript mode (Annotation Mode) Any user can draw on a user's own local display using the same drawing tool as what is seen by the "paint" program. The user can draw a box, a circle, a circle, an ellipse, etc. The user can also eliminate the item on a display.

[0019] This invention can reproduce users' postscript on other displays of all the, therefore can see a display with all the similar users. However, since there are the following factors, those displays may differ.

[0020] (A) If display monitors differ, character, such as expression ability of a color, differs.

[0021] (B) If the protocols (IGA, BGA, etc.) of a display differ, expressions of a graphical image differ and display ability differs.

[0022] (C) GUI differs. Or it is the same. When GUI also differs in a version, display types may differ.

[0023] (D) Some users might change the size of the window where a computer is displayed, and, as a result, the difference may have produced them in the contraction scale.

[0024] These differences make the appearance of the image displayed produce a difference. However, the fundamental contents of all the displays are the same ****. Since the difference in a size is permitted, this invention can be drawn by suitable contraction scale.

[0025] 3. Local postscript mode (Local Annotation Mode) Although the user can put a postscript to a local display, since those postscripts are kept secret, other users cannot see the postscript.

[0026] 4. Observation mode (View Mode) Neither of the users can put a postscript and a command cannot be published. However, action similar to a postscript can be taken. Since the user can move his cursor and other users can see the motion, remote indication (indication of the position of a display thing with cursor) can be performed. Observation mode is useful in the example etc. which specify the mode of the specific user who is, for example during execution in postscript mode as observation mode. In this example, although all the users can put a postscript, "observation" user can only see and cannot put a postscript.

[0027] Explanatory view 3-drawing 14 in each mode illustrates various modes through use of a computer program as an example.

[0028] The drawing 3 host runs an application program.

[0029] The mode is a "postscript."

[0030] A user's input is performed with a host computer.

[0031] A user tries the operation of a computer.

[0032] It is assumed that the user of a host computer tries addition of two numbers using a computer. The first number that is the target of the trial is considered.

[0033] A host user moves a host's cursor on the key of a computer, and a keystroke is tried by clicking a mouse. However, the click of a mouse does not reach an application program. It is because this invention prevents it. An application program does not answer. It is because a program does not receive the click of a mouse.

[0034] That is, if it explains still in detail, GUI will detect movement of this mouse and will generate the "mouse message" which should be generated. GUI sets this mouse message at cue (queue). A mouse message stands by processing in this cue. "The input router (INPUT ROUTER)" shown in drawing 15 reads these messages. Since "postscript mode" is performing now, an input router sends this message to a postscript block. Application does not receive this message, therefore does not answer. This mouse click is disregarded.

[0035] The response "postscript" of a "postscript" can be constituted so that two kinds (or more than it) of responses may be carried out to the message of a mouse. In the one composition, a postscript requires that a mouse should choose a postscript tool first. If selection is not performed at all, a "postscript" will disregard a mouse message.

[0036] Selection is performed by clicking a mouse on a tool image so that it may usually be carried out by a "paint" program. A "postscript" recognizes selection of this tool and, subsequently treats the click of the mouse after it as data for drawing with a selection tool. For example, when a rectangle tool is chosen, the click of the following two mice becomes final and conclusive the corner on a rectangular diagonal line. (Drawing 4 about which it argues later illustrates rectangular drawing.) In the second composition, when it is in postscript mode, a default tool like a pen is chosen automatically. With this composition, if a user performs the trial which pushes the button of a computer (it clicks), drawing of the line of the user using that pen will start (** which is not meant). The user who has recognized this can make drawing of a line finish using either of the publicly known methods.

[0037] So, postscript In the mode, since the (b "postscript") tool is not answered or chosen as the input of a mouse by starting the "postscript" of the (a) default, this

invention carries out the response of ***** which disregards the input of a mouse. The keyboard input from a user is treated similarly. It can also design so that a "postscript" may carry out other responses, of course.

[0038] Pursuit each display of cursor shows the cursor which can be spotted with the mouse of relation. This invention can reproduce each cursor on all the displays. Thus, as shown in drawing 3, when using 3 mice, there is three cursor. (However, since it is easy, even free accepts it and it has illustrated it.)

As a result, if a one user moves a mouse, the cursor corresponding to it will exercise on all the displays.

[0039] Generally :each cursor which three cursor can identify identifies the owner by introducing for example, a color, shape, and a label etc.

[0040] The drawing 4 host runs an application program.

[0041] The mode is postscript mode.

[0042] A user's input is performed with a host computer.

[0043] A user tries drawing of a box on a computer.

[0044] This situation is extremely similar with what is shown in drawing 3. However, it differs in that it has tried to draw a postscript instead of having intention of a user pushing a button in drawing 3 in now.

[0045] It is assumed that the user of a host computer draws a box on a computer. (This box is greatly illustrated too much for emphasis.) As for a box, it is desirable that it is inelastic over the computer itself. This invention reproduces the box on a remote computer. (This box is drawn using the postscript tool which is not illustrated).

[0046] Seeing by drawing 15, an input router sends the flow of logic to a "postscript." A "postscript" calls the suitable GDI function for drawing the box. A "postscript" sends a "postscript message" to connection API. This connection API sends a postscript message to a remote person.

[0047] The "postscript" shown in drawing 15 A receives this postscript message. A this "postscript" block expresses the logic performed in each remote computer. This "postscript" calls a suitable GDI function via block GDI.

[0048] Graphical ["GDI"] Device It is a cable address of an interface. The "GDI function" is included in the big program called GDI.EXE. If a GDI function is called, it will draw a specific graphical image like a circle, a box, or a text based on the input which a user inputs after that. Other GDI functions perform other tasks, such as selection of pen width.

[0049] GDIEXE is a product marketed. "Programming of window drawing 3 .1" (Redmond, Washington,; Microsoft) of a "Windows software development kit" available from Microsoft Corp. in the technical details about GDI.EXE, and the Charles PETOZORUDO work [and] It is indicated to issue in 3 or ISBN 1-55615-395-1992.

[0050] The drawing 5 host runs an application program.

[0051] The mode is "application."

[0052] A user input is performed with a host computer.

[0053] A user tries use of a computer.

[0054] A host user moves cursor on the key "3" of a computer, and clicks a mouse. GUI generates a mouse message and sets it at cue. This invention calls a mouse message and sends the message to an application program (namely, computer program). This program carries out that the (1) key "3" was pressed and the response of drawing the number "3" on the display of a computer using the call of (2) GDI.

[0055] This application program records again the fact that the user inputted "3" in order to perform internal operation of that program.

[0056] When having drawn "3" in a computer, and when this invention displays the button of "3" pushed, it interrupts the GDI call which an application program issues. Other computers reproduce a host display by performing the same GDI function. The paragraph after entitling "general consideration" describes the explanation detailed one layer about this GDI discontinuation.

[0057] Thus, all the users can see a host's user operate a computer simultaneously. (Since very short delay involves, this host's operation is not strictly instantaneous.) But even if a host and a remote computer are located in a line and it operates, probably people cannot notice the delay. An input router recognizes that an application program should point to a mouse message in drawing 15. It points to the flow of logic in application (namely, computer program). application draws the key of "3" (1) pushed -- (2) -- a suitable GDI function is called -- especially therefore, the number "3" is written in in the display of a computer.

[0058] However, this invention catches the GDI call of an application program via GDI capture shown in drawing 15, before they are performed (capture). This invention is attached to the caught call (call), and performs 2 things. As one of them, this invention gives notice of these calls to other computers via a "connection API" block. This action reaches the "GDI display caught" block shown in drawing 15 A. When it does so, this block makes each remote person perform the same GDI function so that block GDI may show.

[0059] As the 2, this invention allows performing the GDI function which the application program which should be executed by a host calls by a host via block GDI shown in drawing 15.

[0060] So, this invention catches the call (call) of the GDI function which the application programm performed. As a result of this invention's giving notice of the caught call to a remote computer, the remote computer can reproduce those calls. This invention is allowed to perform on a host, as the caught call was meant.

[0061] The drawing 6 host runs an application program.

[0062] The mode is "a local postscript."

[0063] A user input is performed with a host computer.

[0064] It tries that a user makes it calculate to a computer.

[0065] In postscript mode, I will assume that there is no default postscript tool given to the user. Under this assumption, a user moves cursor even to the button of a computer and "it pushing" and the "input router" of drawing 15 pass a mouse message for a button to a "postscript" block. This mouse click is that which is not an effective postscript input sequence and which is not then (no tools are chosen), and does not draw "postscript" what.

[0066] A remote computer does not show movement of the cursor corresponding to the mouse of a host computer. It is because the line 5 of drawing 15 does not send a postscript message to other computers during execution of "a local postscript" as mentioned above.

[0067] Even if it pushes the button of a computer on a host display, it does not carry out redrawing the button answered and pushed on it again. It is because application did not receive a mouse message. It is drawing of the pushed computer button that application has responsibility.

[0068] When the default of the postscript is specified as the user who has taken local postscript mode, drawing by the tool is started by the user's mouse click. A user will end drawing by a publicly known method, when a user notices an error.

[0069] The drawing 7 host runs an application program.

[0070] The mode is "a local postscript."

[0071] A user input is performed with a host computer.

[0072] A user tries a postscript to a computer.

[0073] The input router shown in the A section of drawing 15 under these conditions recognizes effective postscripts, such as drawing of a box. An input router points to the flow of the logic in a "postscript" block, and calls the suitable GDI function for postscripts, such as drawing etc. of the box which a "postscript" block shows to drawing 7.

[0074] However, since the postscript is local, as shown in drawing 7, no box is drawn on a remote computer. No data is sent in accordance with the data path 5 of drawing 15.

[0075] The drawing 8 host runs an application program.

[0076] The mode is "observation."

[0077] A user input is performed with a host computer.

[0078] A user tries the operation of a computer.

[0079] As drawing 8 shows, the click of a mouse is disregarded, and nothing occurs in a remote computer.

[0080] Although an input router reads this mouse message in drawing 15, it is prevented so that application may not be reached. It is because the present mode is "observation mode (View Mode)."

[0081] The drawing 9 host runs an application program.

[0082] The mode is a "postscript."

[0083] A remote computer performs a user input.

[0084] A user tries the operation of a computer.

[0085] A user moves a mouse cursor on the button of a computer, and it is assumed that a mouse is clicked. This click is disregarded. Although other computers (a host and other remote computers) show movement of a user's cursor, nothing shows them but it. This is because no tools are chosen.

[0086] In drawing 15 A, an input router prevents that this mouse message reaches application. A logical operation is applied to a "postscript" and a "postscript" draws cursor on a user's remote display via a GDI block. A "postscript" sends data to connection API and this connection API applies a logical operation to the "postscript" of drawing 15. this "postscript" expresses the contents of postscript logic which exist on other two computers (namely, a host -- and -- being concerned -- others -- a remote computer). A these "postscript" block draws at each place via the GDI block which shows drawing 15 the cursor corresponding to each user's cursor. GDI of drawing 15 expresses the contents

of the GDI function call.

[0087] While a host can use one tool like a box draughting tool, the remote computer can use another tool like the tool describing a circle.

[0088] The drawing 10 host runs an application program.

[0089] The mode is a "postscript."

[0090] A remote computer performs a user input.

[0091] A user tries the postscript to a computer.

[0092] It is assumed that a postscript is a box. A box is drawn on all the displays. In drawing 15 A, the input router in a user's remote computer sends a mouse message to a "postscript" block. A "postscript" performs two things. One of them calls suitable GDI for a "postscript" to draw the box which is a postscript.

[0093] The 2, a "postscript" sends a postscript message to connection API, and connection API sends a postscript message to the computer of another side. However, these one side is host computers and another side is a remote computer. A host's logic process reaches the "postscript" of drawing 15, and the "postscript" which has a logic process of the remote computer of said another side in drawing 15 A is reached.

[0094] Both call a suitable GDI function and a these "postscript" block makes the postscript of a user desire draw on this. However, a host's logical path 5 is not taken up at this time. It is because it is not necessary to reproduce a host's postscript to other computers.

[0095] The drawing 11 host runs an application program.

[0096] The mode is application.

[0097] A remote computer performs a user input.

[0098] A user tries the operation of a computer.

[0099] Readers need to recollect that this computer program is loaded only to the host. However, it hopes that the user of a remote computer wants to operate [it].

[0100] The input router of the remote user who shows drawing 15 A sends a mouse message to connection API. A host receives these messages and these messages are sent to the input router of the host who shows drawing 15. As for a host's input router, block application sends these messages (to namely, application program which is a computer program). This program performs two important things.

[0101] a remote mouse -- them -- in spite of having carried out message appearance, the computer program treats those messages as if it was published with the host's mouse. A computer program will answer by the usual method, if it gets to know that the user inputted data (namely, "3"). The response should show the computer button "3" :
[Containing the following thing] (1) Pushed. (2) Write in the number "3" in the display of a computer. And perform internal calculation of (3) itself.

[0102] However, before a computer program can perform now (1) of the foregoing paragraph, and (2), this invention catches the GDI function which a computer program calls first. This prehension is illustrated by the GDI prehension block of drawing 15. In the case of this prehension, this invention performs two things. This invention sends one of them of these GDI functions to connection API (in order to be able to use other computers). In a user's remote computer, connection API shown in drawing 15 A sends this GDI function to "the caught GDI display." This display creates the duplicate of a host display. This invention performs the GDI function which should be performed by a host (going via block GDI shown in drawing 15) by a host as the 2. So, the general sequence of a phenomenon carries out the trial with which : and the remote user who is as follows push a computer button.

[0103] - This invention which is operating on a remote computer detects this trial, and sends data to the computer program which is running on the host. This data takes the form of a message. The program of a computer "thinks" that this message came from the host's mouse.

[0104] - A computer program usually works to a passage and draws an image on a host display via a GDI call.

[0105] - This invention catches a GDI call and notify them to a remote computer.

[0106] - A remote computer reproduces a host's window. The remote user can do in this way and can operate remotely the computer program which is running on the host.

[0107] If it summarizes by another way of speaking, this invention will generate a mouse message to a host based on the mouse message of a remote computer. (It is running on the host) A computer program answers a mouse message as if those mouse messages were generated in the host. The GDI call which the computer program performed is interrupted for this invention, and the same GDI call in a remote computer is performed by it.

Therefore, a host display is reproduced to a remote computer.

[0108] The drawing 12 host runs an application program.

[0109] The mode is "a local postscript."

[0110] A remote computer performs a user input.

[0111] A user tries OPERESSHON of a computer.

[0112] The mouse click of the user concerned is disregarded. Since selection of a tool has gone wrong, movement of this mouse is answered and nothing appears on other displays.

[0113] The drawing 13 host runs an application program.

[0114] The mode is "a local postscript."

[0115] A user input is performed by the remote computer.

[0116] A user tries the postscript to a computer.

[0117] As described above, a postscript is drawn on a user's display. No postscript occurs on other displays.

[0118] The drawing 14 host runs an application program.

[0119] The mode is "observation."

[0120] A remote computer performs a user input.

[0121] A user tries the operation of a computer.

[0122] A mouse click is disregarded although the cursor of a mouse moves on a user's display. Other two displays do not show movement of a user's mouse cursor.

[0123] The program of general consideration 1. another ** draws in a different portion of the whole display. A display is drawn using a GDI function. However, the portion from which at least one display differs is drawn by a different program, respectively.

[0124] In spite of performing these drawing [all] operations using a GDI function, GDI functions are not the exclusive display duplicate communication media between computers.

[0125] A postscript includes the data transfer between computers of one gestalt.

[0126] Drawing by an application program contains the thing of other gestalten.

[0127] For example, when a user puts a postscript, a user's mouse message is reproduced as a "message" in other computers via the course 5 shown in drawing 15. These duplicate messages rank second and make a suitable GDI call for an each (it can set to other computers) "postscript" block to draw the postscript publish. That is, a GDI call is not

directly sent to other computers from the user who is performing the postscript.

[0128] When the application program is making the graphical image draw on a display, this invention interrupts the GDI call (going via "GDI prehension" shown in drawing 15), and makes the GDI call reproduce on other computers in contrast with it.

[0129] The main reasons for performing a 2 reasons [for a difference] different procedure (the duplicate of a mouse message and the duplicate of a GDI call) are because a postscript is stored in the position from which display information differed.

[0130] That is, it returns to the computer of drawing 2 and an application program stores the image of a computer by the following general methods. Postscript data is stored by this invention. Application program data is stored by an application program (a host has). The image of each key is stored as data in which a GDI function can pull out the key. This data includes information, including a position, a size, a color, etc. Each key contains the number of relation. The number is storable as a text character with information, including a position, a size, font types, etc.

[0131] Although postscript data is stored in a different place, it is stored by the same general method.

[0132] When any of a postscript or an application program they are needs a bit map, a bit map is stored by GUI by a publicly known method.

[0133] This invention combines a postscript image with the image of application by the publicly known art about masking. That is, this invention reproduces the GDI function which the remote computer received to a bit map. This invention reproduces the received postscript information to a different bit map from it. Mask credit of these two bit maps is carried out mutually.

[0134] For the purpose of a user enabling it to save an application image without a postscript etc., it dissociates from application data and postscript data is saved. As this substitute, the user can save postscript data in the form of a display put a postscript independently.

[0135] Another reason for carrying out separate preservation is convenient [saving postscript data separately] to draw the display which does not have postscript data at all. When postscript data has mixed with computer image data, it is difficult to remove postscript data also until it is not impossible.

[0136] supposing it transmits a GDI call exclusively, and reproduction of a message is not performed at all namely,, excessive work with being separate for constituting so that it may store will be needed in postscript data.

[0137] 2. He can understand discontinuation of GDI, or discontinuation of prehension GDI as follows.

[0138] A. At the time of start-up, this invention replaces the first 5 bytes of each GDI function by a certain specific program (JUMP), i.e., the jump instructions to Trap.GDI.

[0139] B. Trap.GDI gains the parameter to a desired graphical image (it is a position of the angle of 2 diagonal lines in the case of a box, etc.). And PkgDispCall which is a sub program is called. Trap.GDI also replaces 5 bytes of the beginning again.

[0140] C. Receive a parameter from PkgDispCall and Trap.GDI and generate a BUJIEKUTO structure (object structure). This object structure is the data block included all the elements required for other computers to draw a box.

[0141] For example, the object structure includes the information about the size and position of a box. GUI draws an image in "context (context)." This context contains the thing of the pen which has others width and a color, and characteristic. This invention pursues the context of each computer. If the context of the drawn box differs from the context of the remote computer, PkgDispCall will include data required for other computers to generate a right context.

[0142] D. an object structure is shipped by other computers -- being concerned -- others - - perform the GDI function that a computer is the same.

[0143] E. This invention performs an original GDI function.

[0144] 3. The whole display is not transmitted. A display is not replaced by bitwise. For example, the image of the computer shown in drawing 2 can also transmit between computers by bitwise. If a computer occupies 200x300-pixel space, the information about 60,000 pixels (namely, pixel of 200x300) must be sent. As this substitute, the specific computer shown in drawing 2 is treated as 18 rectangles and 1 text sentence character to 18 rectangles of each, and, as a result, becomes 34 objects in all. Each object needs a size and a parameter like a position. The number of parameters is small and is 3 thru/or 10. 340 data must be sent if it assumes that there are ten parameters. Of course, although it is dependent on various factors, if a small number of bytes is assumed to each piece, it is sufficient for the size of each data piece (data pieces).

[0145] So, this invention reduces required data to a maximum of 340 piece for an object duplicate 60,000 piece with the duplicate of bitwise. Of course depending on an object, the gestalt of a bit map may be taken, and it must send for every bit. However, generally a bit map does not appear rarely. When sending a bit map generally, it is expected that it is sent only once.

[0146] Furthermore, object data is possible and **** compression is carried out. That is, signal transduction between computers will be performed in the form of compressed data, if possible. Compression is publicly known at this technical field.

[0147] 4. The communication between the gestalt computers of a data link can take some gestalten. The network, local-area, and wide area which are marketed can be used. The

ISDN telephone line marketed is provided from the local telecommunications company, and can also use these. Modem communication can also be used.

[0148] 5 The package of marketing which detects the message which answered the message sensing input device of the advanced technology, and was generated in GUI can be used. One of such the packages is WINSIGHT, and this is Borland. It can obtain from International. However, these packages are considered not to carry out a message report at a remote computer.

[0149] 6 "GDI prehension" which becomes instead of
The method of replacing with prehension of the above-mentioned graphics is as follows. Before calling actual system GDI for GDI which this system gives, it replaces by the separate procedure which processes a GDI call. The name of system GDI is changed in order to prevent mixing up by two inter modules. In order to catch the GDI call performed via the module with which the system is provided, the same technique also as USR.EXE is used.

[0150] 7. -- even -- the above computer can run an application program. A certain specific computer can act as a host to one program, and a remote computer can be acted to another program. For example, one computer can run a word processor and another computer can run a paint program. To each program, each is a host.

[0151] Since the software of this invention carried in each computer is the same or the same in a parenchyma top, it can run, as all the users mentioned above the word processor or the CAD program.

[0152] 8. There is "cursor" of the type of "real" cursor and two "false" cursor. Each GUI owns the "real (it real(s) and is actual)" cursor of itself. It is not generated that it is a GUI function but actual cursor is generated with the independent function in GUI. It can be considered that this cursor is a bit map to which you answer movement of a mouse and GUI makes it move. Adding to the real cursor controlled by a local mouse, this invention generates the "false" cursor to the participant in each remoteness. A pseudo-cursor is generated using a GDI function. Real cursor changes with movement of cursor occasionally. For example, when it is on a tool bar, the cursor can take the shape of an arrow, and when it is on a client area, it can change to the form of a hand. This change is often set on the basis of control of an application program.

[0153] So, when a remote user (it is shown in drawing 11 like) controls the application program which is running on the host machine, an application program can change the cursor on a host machine, without using a GDI call. Therefore, prehension of GDI shown in drawing 15 and 15A is inefficiency to reproducing the cursor which changed on a remote display.

[0154] In order to cope with this problem, this invention supervises the function to change real cursor (for example, the SetCursor command etc.). This invention reproduces cursor change on a remote computer.

[0155] It is also one method to execute the same SetCursor command. The method instead of another will be changing remote cursor, when host cursor's changes, or making a bit map draw by performing the sequence of a suitable GDI call.

[0156] 9. The whole display is not reproduced. This invention only reproduces the window which the user of a display identifies. That is, the user can hold a piece of drawing like Note Pad in secret in a period of a meeting. A GDI call uses the task handle (taskhandle). A GDI call will not be shared if a task handle does not refer to the item to share.

[0157] [Effect]In two or more computers by which interconnection was carried out, this invention by the observation mode, postscript mode, and control mode which were controlled by the program of the host computer. Each user of the computer of these plurality participates in control of the program of a host computer at remoteness, Or a postscript can be sent to the computer of the user of self or others, it can perform observing other users' action, and communication of meeting resemblance can be performed among these computer users.

TECHNICAL FIELD

[Industrial Application]This invention relates to the system by which two or more users can make it enable it to operate a single computer program by remote control. This invention generates the common visual image distributed to all the computers. The user can create a postscript on a common display (display). If a user creates a postscript, this invention will reproduce the postscript (annotation) on all the displays. A postscript can also be individually held secretly by a user.

PRIOR ART

[Description of the Prior Art]A modernistic phone system enables two or more parties which are in a different place to hold a conference. However, the conference call can give no facilities obtained at the meeting in which all participants meet on the common table of a conference room.

[0003] For example, in a conference room, the participant can see a drawing or a product in connection with subject, etc. Such observation is impossible in a conference call.

[0004] This invention gives the system which enables it to obtain when [expedient/many of] the participant has participated in the meeting physically in the remote place.

EFFECT OF THE INVENTION

[Effect] In two or more computers by which interconnection was carried out, this invention by the observation mode, postscript mode, and control mode which were controlled by the program of the host computer. Each user of the computer of these plurality participates in control of the program of a host computer at remoteness, Or a postscript can be sent to the computer of the user of self or others, it can perform observing other users' action, and communication of meeting resemblance can be performed among these computer users.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention]The technical problem of this invention is giving the improved electronic meeting system.

[0006] Another technical problem of this invention is giving the system by which it enables it for a user to operate a computer program remotely.

[0007] Another technical problem of this invention is giving the system by which it enables it for those computers to operate the single program which exists in one of two or more of the computers.

[0008] Another technical problem of this invention is giving the system to which it can be added while the user of two or more computers can see a common display.

MEANS

[Means for Solving the Problem]This invention connects a multiplex computer mutually as the one gestalt. A single computer runs an application program. A display created by the program is reproduced on a display of other computers of all the. The user of each computer can put a postscript on his display. [who is using it] This invention can reproduce those postscripts on other displays of all the. This invention can run a program by which a user of one computer is loaded on others' computer.

EXAMPLE

[Example]

General-view drawing 1 shows three computers connected with the telephone wire. Each computer runs the multitasking graphical user interface (Graphical User Interface, GUI) by which a message drive is carried out. The example has a thing named the Windows marketed from Microsoft Corporation of Redmond, Washington. It is such. GUI is also called operating operating environment (operating environments) again.

[0011] the user of GUI gives an interaction via the program using Windows -- it can do. This invention reproduces not the whole display of the computer which exists far away

but the window of shoes to choose. A user becomes possible [maintaining individual area on its own display (one's area which is not shared)] by this alternative duplicate. Each computer can run the software which this invention person developed again. One computer (host) runs an application program. (Since it has the multitasking ability of GUI, a host It is possible to run both programs).

[0012] This invention has the operation of four dominant modes.

[0013] 1. Application mode (Application Mode)

Any user of three computers shown in drawing 1 can give a command to an application program. For example, an application program will assume that it is what simulates a stock type computer. The first state is shown in drawing 2. In this case, each computer display illustrates a computer. The user of : and the host who assumes that the following phenomenon happens pushes the button "3" of a computer (it responds to the design of a computer program and they are an input of a keyboard, or an input of a mouse). This is answered and each computer displays "3" on the viewing area.

[0014]- One of remote users pushes "+." Other remote users push "6."

[0015] - A host user pushes "=".

[0016] At this time, all the computers display "9" which is the sum of 3 and 6. Users perform operation of this computer program in a group, and each one of displays display that result.

[0017] This computer program does not take into consideration whether it did not take into consideration whether which user pushed the button, and someone of users pushed the button, as long as a lawful sequence is received. (However, it is assumed that users are cooperative and neither of the users does nonfeasance of the operation of a computer).

[0018] 2. Postscript mode (Annotation Mode) Any user can draw on a user's own local display using the same drawing tool as what is seen by the "paint" program. The user can draw a box, a circle, a circle, an ellipse, etc. The user can also eliminate the item on a display.

[0019] This invention can reproduce users' postscript on other displays of all the, therefore can see a display with all the similar users. However, since there are the following factors, those displays may differ.

[0020] (A) If display monitors differ, character, such as expression ability of a color, differs.

[0021] (B) If the protocols (IGA, BGA, etc.) of a display differ, expressions of a graphical image differ and display ability differs.

[0022] (C) GUI differs. Or it is the same. When GUI also differs in a version, display

types may differ.

[0023] (D) Some users might change the size of the window where a computer is displayed, and, as a result, the difference may have produced them in the contraction scale.

[0024] These differences make the appearance of the image displayed produce a difference. However, the fundamental contents of all the displays are the same ****. Since the difference in a size is permitted, this invention can be drawn by suitable contraction scale.

[0025] 3. Local postscript mode (Local Annotation Mode) Although the user can put a postscript to a local display, since those postscripts are kept secret, other users cannot see the postscript.

[0026] 4. Observation mode (View Mode) Neither of the users can put a postscript and a command cannot be published. However, action similar to a postscript can be taken. Since the user can move his cursor and other users can see the motion, remote indication (indication of the position of a display thing with cursor) can be performed. Observation mode is useful in the example etc. which specify the mode of the specific user who is, for example during execution in postscript mode as observation mode. In this example, although all the users can put a postscript, "observation" user can only see and cannot put a postscript.

[0027] Explanatory view 3-drawing 14 in each mode illustrates various modes through use of a computer program as an example.

[0028] The drawing 3 host runs an application program.

[0029] The mode is a "postscript."

[0030] A user's input is performed with a host computer.

[0031] A user tries the operation of a computer.

[0032] It is assumed that the user of a host computer tries addition of two numbers using a computer. The first number that is the target of the trial is considered.

[0033] A host user moves a host's cursor on the key of a computer, and a keystroke is tried by clicking a mouse. However, the click of a mouse does not reach an application program. It is because this invention prevents it. An application program does not answer. It is because a program does not receive the click of a mouse.

[0034] That is, if it explains still in detail, GUI will detect movement of this mouse and will generate the "mouse message" which should be generated. GUI sets this mouse message at cue (queue). A mouse message stands by processing in this cue. "The input

router (INPUT ROUTER)" shown in drawing 15 reads these messages. Since "postscript mode" is performing now, an input router sends this message to a postscript block. Application does not receive this message, therefore does not answer. This mouse click is disregarded.

[0035]The response "postscript" of a "postscript" can be constituted so that two kinds (or more than it) of responses may be carried out to the message of a mouse. In the one composition, a postscript requires that a mouse should choose a postscript tool first. If selection is not performed at all, a "postscript" will disregard a mouse message.

[0036]Selection is performed by clicking a mouse on a tool image so that it may usually be carried out by a "paint" program. A "postscript" recognizes selection of this tool and, subsequently treats the click of the mouse after it as data for drawing with a selection tool. For example, when a rectangle tool is chosen, the click of the following two mice becomes final and conclusive the corner on a rectangular diagonal line. (Drawing 4 about which it argues later illustrates rectangular drawing.) In the second composition, when it is in postscript mode, a default tool like a pen is chosen automatically. With this composition, if a user performs the trial which pushes the button of a computer (it clicks), drawing of the line of the user using that pen will start (** which is not meant). The user who has recognized this can make drawing of a line finish using either of the publicly known methods.

[0037]So, postscript In the mode, since the (b "postscript") tool is not answered or chosen as the input of a mouse by starting the "postscript" of the (a) default, this invention carries out the response of ***** which disregards the input of a mouse. The keyboard input from a user is treated similarly. It can also design so that a "postscript" may carry out other responses, of course.

[0038]Pursuit each display of cursor shows the cursor which can be spotted with the mouse of relation. This invention can reproduce each cursor on all the displays. Thus, as shown in drawing 3, when using 3 mice, there is three cursor. (However, since it is easy, even free accepts it and it has illustrated it.)

As a result, if a one user moves a mouse, the cursor corresponding to it will exercise on all the displays.

[0039]Generally :each cursor which three cursor can identify identifies the owner by introducing for example, a color, shape, and a label etc.

[0040]The drawing 4 host runs an application program.

[0041]The mode is postscript mode.

[0042]A user's input is performed with a host computer.

[0043]A user tries drawing of a box on a computer.

[0044]This situation is extremely similar with what is shown in drawing 3. However, it differs in that it has tried to draw a postscript instead of having intention of a user pushing a button in drawing 3 in now.

[0045]It is assumed that the user of a host computer draws a box on a computer. (This box is greatly illustrated too much for emphasis.) As for a box, it is desirable that it is inelastic over the computer itself. This invention reproduces the box on a remote computer. (This box is drawn using the postscript tool which is not illustrated).

[0046]Seeing by drawing 15, an input router sends the flow of logic to a "postscript." A "postscript" calls the suitable GDI function for drawing the box. A "postscript" sends a "postscript message" to connection API. This connection API sends a postscript message to a remote person.

[0047]The "postscript" shown in drawing 15 A receives this postscript message. A this "postscript" block expresses the logic performed in each remote computer. This "postscript" calls a suitable GDI function via block GDI.

[0048]Graphical ["GDI"] Device It is a cable address of an interface. The "GDI function" is included in the big program called GDI.EXE. If a GDI function is called, it will draw a specific graphical image like a circle, a box, or a text based on the input which a user inputs after that. Other GDI functions perform other tasks, such as selection of pen width.

[0049]GDIEXE is a product marketed. "Programming of window drawing 3 .1" (Redmond, Washington,; Microsoft) of a "Windows software development kit" available from Microsoft Corp. in the technical details about GDI.EXE, and the Charles PETOZORUDO work [and] It is indicated to issue in 3 or ISBN 1-55615-395-1992.

[0050]The drawing 5 host runs an application program.

[0051]The mode is "application."

[0052]A user input is performed with a host computer.

[0053]A user tries use of a computer.

[0054]A host user moves cursor on the key "3" of a computer, and clicks a mouse. GUI generates a mouse message and sets it at cue. This invention calls a mouse message and sends the message to an application program (namely, computer program). This program carries out that the (1) key "3" was pressed and the response of drawing the number "3" on the display of a computer using the call of (2) GDI.

[0055]This application program records again the fact that the user inputted "3" in order to perform internal operation of that program.

[0056]When having drawn "3" in a computer, and when this invention displays the button of "3" pushed, it interrupts the GDI call which an application program issues. Other computers reproduce a host display by performing the same GDI function. The paragraph after entitling "general consideration" describes the explanation detailed one layer about this GDI discontinuation.

[0057]Thus, all the users can see a host's user operate a computer simultaneously. (Since very short delay involves, this host's operation is not strictly instantaneous.) But even if a host and a remote computer are located in a line and it operates, probably people cannot notice the delay. An input router recognizes that an application program should point to a mouse message in drawing 15. It points to the flow of logic in application (namely, computer program). application draws the key of "3" (1) pushed -- (2) -- a suitable GDI function is called -- especially therefore, the number "3" is written in in the display of a computer.

[0058]However, this invention catches the GDI call of an application program via GDI capture shown in drawing 15, before they are performed (capture). This invention is attached to the caught call (call), and performs 2 things. As one of them, this invention gives notice of these calls to other computers via a "connection API" block. This action reaches the "GDI display caught" block shown in drawing 15 A. When it does so, this block makes each remote person perform the same GDI function so that block GDI may show.

[0059]As the 2, this invention allows performing the GDI function which the application program which should be executed by a host calls by a host via block GDI shown in drawing 15.

[0060]So, this invention catches the call (call) of the GDI function which the application program performed. As a result of this invention's giving notice of the caught call to a remote computer, the remote computer can reproduce those calls. This invention is allowed to perform on a host, as the caught call was meant.

[0061]The drawing 6 host runs an application program.

[0062]The mode is "a local postscript."

[0063]A user input is performed with a host computer.

[0064]It tries that a user makes it calculate to a computer.

[0065]In postscript mode, I will assume that there is no default postscript tool given to the user. Under this assumption, a user moves cursor even to the button of a computer and "it pushing" and the "input router" of drawing 15 pass a mouse message for a button to a "postscript" block. This mouse click is that which is not an effective postscript input sequence and which is not then (no tools are chosen), and does not draw "postscript"

what.

[0066]A remote computer does not show movement of the cursor corresponding to the mouse of a host computer. It is because the line 5 of drawing 15 does not send a postscript message to other computers during execution of "a local postscript" as mentioned above.

[0067]Even if it pushes the button of a computer on a host display, it does not carry out redrawing the button answered and pushed on it again. It is because application did not receive a mouse message. It is drawing of the pushed computer button that application has responsibility.

[0068]When the default of the postscript is specified as the user who has taken local postscript mode, drawing by the tool is started by the user's mouse click. A user will end drawing by a publicly known method, when a user notices an error.

[0069]The drawing 7 host runs an application program.

[0070]The mode is "a local postscript."

[0071]A user input is performed with a host computer.

[0072]A user tries a postscript to a computer.

[0073]The input router shown in the A section of drawing 15 under these conditions recognizes effective postscripts, such as drawing of a box. An input router points to the flow of the logic in a "postscript" block, and calls the suitable GDI function for postscripts, such as drawing etc. of the box which a "postscript" block shows to drawing 7.

[0074]However, since the postscript is local, as shown in drawing 7, no box is drawn on a remote computer. No data is sent in accordance with the data path 5 of drawing 15.

[0075]The drawing 8 host runs an application program.

[0076]The mode is "observation."

[0077]A user input is performed with a host computer.

[0078]A user tries the operation of a computer.

[0079]As drawing 8 shows, the click of a mouse is disregarded, and nothing occurs in a remote computer.

[0080]Although an input router reads this mouse message in drawing 15, it is prevented so that application may not be reached. It is because the present mode is "observation

mode (View Mode)."

[0081]The drawing 9 host runs an application program.

[0082]The mode is a "postscript."

[0083]A remote computer performs a user input.

[0084]A user tries the operation of a computer.

[0085]A user moves a mouse cursor on the button of a computer, and it is assumed that a mouse is clicked. This click is disregarded. Although other computers (a host and other remote computers) show movement of a user's cursor, nothing shows them but it. This is because no tools are chosen.

[0086]In drawing 15 A, an input router prevents that this mouse message reaches application. A logical operation is applied to a "postscript" and a "postscript" draws cursor on a user's remote display via a GDI block. A "postscript" sends data to connection API and this connection API applies a logical operation to the "postscript" of drawing 15. this "postscript" expresses the contents of postscript logic which exist on other two computers (namely, a host -- and -- being concerned -- others -- a remote computer). A these "postscript" block draws at each place via the GDI block which shows drawing 15 the cursor corresponding to each user's cursor. GDI of drawing 15 expresses the contents of the GDI function call.

[0087]While a host can use one tool like a box draughting tool, the remote computer can use another tool like the tool describing a circle.

[0088]The drawing 10 host runs an application program.

[0089]The mode is a "postscript."

[0090]A remote computer performs a user input.

[0091]A user tries the postscript to a computer.

[0092]It is assumed that a postscript is a box. A box is drawn on all the displays. In drawing 15 A, the input router in a user's remote computer sends a mouse message to a "postscript" block. A "postscript" performs two things. One of them calls suitable GDI for a "postscript" to draw the box which is a postscript.

[0093]The 2, a "postscript" sends a postscript message to connection API, and connection API sends a postscript message to the computer of another side. However, these one side is host computers and another side is a remote computer. A host's logic process reaches the "postscript" of drawing 15, and the "postscript" which has a logic process of the remote computer of said another side in drawing 15 A is reached.

[0094]Both call a suitable GDI function and a these "postscript" block makes the postscript of a user desire draw on this. However, a host's logical path 5 is not taken up at this time. It is because it is not necessary to reproduce a host's postscript to other computers.

[0095]The drawing 11 host runs an application program.

[0096]The mode is application.

[0097]A remote computer performs a user input.

[0098]A user tries the operation of a computer.

[0099]Readers need to recollect that this computer program is loaded only to the host. However, it hopes that the user of a remote computer wants to operate [it].

[0100]The input router of the remote user who shows drawing 15 A sends a mouse message to connection API. A host receives these messages and these messages are sent to the input router of the host who shows drawing 15. As for a host's input router, block application sends these messages (to namely, application program which is a computer program). This program performs two important things.

[0101]a remote mouse -- them -- in spite of having carried out message appearance, the computer program treats those messages as if it was published with the host's mouse. A computer program will answer by the usual method, if it gets to know that the user inputted data (namely, "3"). The response should show the computer button "3" : [Containing the following thing] (1) Pushed. (2) Write in the number "3" in the display of a computer. And perform internal calculation of (3) itself.

[0102]However, before a computer program can perform now (1) of the foregoing paragraph, and (2), this invention catches the GDI function which a computer program calls first. This prehension is illustrated by the GDI prehension block of drawing 15. In the case of this prehension, this invention performs two things. This invention sends one of them of these GDI functions to connection API (in order to be able to use other computers). In a user's remote computer, connection API shown in drawing 15 A sends this GDI function to "the caught GDI display." This display creates the duplicate of a host display. This invention performs the GDI function which should be performed by a host (going via block GDI shown in drawing 15) by a host as the 2. So, the general sequence of a phenomenon carries out the trial with which : and the remote user who is as follows push a computer button.

[0103]- This invention which is operating on a remote computer detects this trial, and sends data to the computer program which is running on the host. This data takes the form of a message. The program of a computer "thinks" that this message came from the

host's mouse.

[0104]- A computer program usually works to a passage and draws an image on a host display via a GDI call.

[0105]- This invention catches a GDI call and notify them to a remote computer.

[0106]- A remote computer reproduces a host's window. The remote user can do in this way and can operate remotely the computer program which is running on the host.

[0107]If it summarizes by another way of speaking, this invention will generate a mouse message to a host based on the mouse message of a remote computer. (It is running on the host) A computer program answers a mouse message as if those mouse messages were generated in the host. The GDI call which the computer program performed is interrupted for this invention, and the same GDI call in a remote computer is performed by it.

Therefore, a host display is reproduced to a remote computer.

[0108]The drawing 12 host runs an application program.

[0109]The mode is "a local postscript."

[0110]A remote computer performs a user input.

[0111]A user tries OPERESSHON of a computer.

[0112]The mouse click of the user concerned is disregarded. Since selection of a tool has gone wrong, movement of this mouse is answered and nothing appears on other displays.

[0113]The drawing 13 host runs an application program.

[0114]The mode is "a local postscript."

[0115]A user input is performed by the remote computer.

[0116]A user tries the postscript to a computer.

[0117]As described above, a postscript is drawn on a user's display. No postscript occurs on other displays.

[0118]The drawing 14 host runs an application program.

[0119]The mode is "observation."

[0120]A remote computer performs a user input.

[0121]A user tries the operation of a computer.

[0122]A mouse click is disregarded although the cursor of a mouse moves on a user's display. Other two displays do not show movement of a user's mouse cursor.

[0123]The program of general consideration 1. another ** draws in a different portion of the whole display. A display is drawn using a GDI function. However, the portion from which at least one display differs is drawn by a different program, respectively.

[0124]In spite of performing these drawing [all] operations using a GDI function, GDI functions are not the exclusive display duplicate communication media between computers.

[0125]A postscript includes the data transfer between computers of one gestalt.

[0126]Drawing by an application program contains the thing of other gestalten.

[0127]For example, when a user puts a postscript, a user's mouse message is reproduced as a "message" in other computers via the course 5 shown in drawing 15. These duplicate messages rank second and make a suitable GDI call for an each (it can set to other computers) "postscript" block to draw the postscript publish. That is, a GDI call is not directly sent to other computers from the user who is performing the postscript.

[0128]When the application program is making the graphical image draw on a display, this invention interrupts the GDI call (going via "GDI prehension" shown in drawing 15), and makes the GDI call reproduce on other computers in contrast with it.

[0129]The main reasons for performing a 2 reasons [for a difference] different procedure (the duplicate of a mouse message and the duplicate of a GDI call) are because a postscript is stored in the position from which display information differed.

[0130]That is, it returns to the computer of drawing 2 and an application program stores the image of a computer by the following general methods. Postscript data is stored by this invention. Application program data is stored by an application program (a host has). The image of each key is stored as data in which a GDI function can pull out the key. This data includes information, including a position, a size, a color, etc. Each key contains the number of relation. The number is storable as a text character with information, including a position, a size, font types, etc.

[0131]Although postscript data is stored in a different place, it is stored by the same general method.

[0132]When any of a postscript or an application program they are needs a bit map, a bit map is stored by GUI by a publicly known method.

[0133]This invention combines a postscript image with the image of application by the publicly known art about masking. That is, this invention reproduces the GDI function which the remote computer received to a bit map. This invention reproduces the received postscript information to a different bit map from it. Mask credit of these two bit maps is carried out mutually.

[0134]For the purpose of a user enabling it to save an application image without a postscript etc., it dissociates from application data and postscript data is saved. As this substitute, the user can save postscript data in the form of a display put a postscript independently.

[0135]Another reason for carrying out separate preservation is convenient [saving postscript data separately] to draw the display which does not have postscript data at all. When postscript data has mixed with computer image data, it is difficult to remove postscript data also until it is not impossible.

[0136]supposing it transmits a GDI call exclusively, and reproduction of a message is not performed at all namely,, excessive work with being separate for constituting so that it may store will be needed in postscript data.

[0137]2. He can understand discontinuation of GDI, or discontinuation of prehension GDI as follows.

[0138]A. At the time of start-up, this invention replaces the first 5 bytes of each GDI function by a certain specific program (JUMP), i.e., the jump instructions to Trap.GDI.

[0139]B. Trap.GDI gains the parameter to a desired graphical image (it is a position of the angle of 2 diagonal lines in the case of a box, etc.). And PkgDispCall which is a sub program is called. Trap.GDI also replaces 5 bytes of the beginning again.

[0140]C. Receive a parameter from PkgDispCall and Trap.GDI and generate a BUJIEKUTO structure (object structure). This object structure is the data block included all the elements required for other computers to draw a box.

[0141]For example, the object structure includes the information about the size and position of a box. GUI draws an image in "context (context)." This context contains the thing of the pen which has others width and a color, and characteristic. This invention pursues the context of each computer. If the context of the drawn box differs from the context of the remote computer, PkgDispCall will include data required for other computers to generate a right context.

[0142]D. an object structure is shipped by other computers -- being concerned -- others -- perform the GDI function that a computer is the same.

[0143]E. This invention performs an original GDI function.

[0144]3. The whole display is not transmitted. A display is not replaced by bitwise. For example, the image of the computer shown in drawing 2 can also transmit between computers by bitwise. If a computer occupies 200x300-pixel space, the information about 60,000 pixels (namely, pixel of 200x300) must be sent. As this substitute, the specific computer shown in drawing 2 is treated as 18 rectangles and 1 text sentence character to 18 rectangles of each, and, as a result, becomes 34 objects in all. Each object needs a size and a parameter like a position. The number of parameters is small and is 3 thru/or 10. 340 data must be sent if it assumes that there are ten parameters. Of course, although it is dependent on various factors, if a small number of bytes is assumed to each piece, it is sufficient for the size of each data piece (data pieces).

[0145]So, this invention reduces required data to a maximum of 340 piece for an object duplicate 60,000 piece with the duplicate of bitwise. Of course depending on an object, the gestalt of a bit map may be taken, and it must send for every bit. However, generally a bit map does not appear rarely. When sending a bit map generally, it is expected that it is sent only once.

[0146]Furthermore, object data is possible and ***** compression is carried out. That is, signal transduction between computers will be performed in the form of compressed data, if possible. Compression is publicly known at this technical field.

[0147]4. The communication between the gestalt computers of a data link can take some gestalten. The network, local-area, and wide area which are marketed can be used. The ISDN telephone line marketed is provided from the local telecommunications company, and can also use these. Modem communication can also be used.

[0148]5 The package of marketing which detects the message which answered the message sensing input device of the advanced technology, and was generated in GUI can be used. One of such the packages is WINSIGHT, and this is Borland. It can obtain from International. However, these packages are considered not to carry out a message report at a remote computer.

[0149]6 "GDI prehension" which becomes instead of
The method of replacing with prehension of the above-mentioned graphics is as follows. Before calling actual system GDI for GDI which this system gives, it replaces by the separate procedure which processes a GDI call. The name of system GDI is changed in order to prevent mixing up by two inter modules. In order to catch the GDI call performed via the module with which the system is provided, the same technique also as USR.EXE is used.

[0150]7. -- even -- the above computer can run an application program. A certain specific computer can act as a host to one program, and a remote computer can be acted to another program. For example, one computer can run a word processor and another computer can run a paint program. To each program, each is a host.

[0151]Since the software of this invention carried in each computer is the same or the same in a parenchyma top, it can run, as all the users mentioned above the word processor or the CAD program.

[0152]8. There is "cursor" of the type of "real" cursor and two "false" cursor. Each GUI owns the "real (it real(s) and is actual)" cursor of itself. It is not generated that it is a GUI function but actual cursor is generated with the independent function in GUI. It can be considered that this cursor is a bit map to which you answer movement of a mouse and GUI makes it move. Adding to the real cursor controlled by a local mouse, this invention generates the "false" cursor to the participant in each remoteness. A pseudo-cursor is generated using a GDI function. Real cursor changes with movement of cursor occasionally. For example, when it is on a tool bar, the cursor can take the shape of an arrow, and when it is on a client area, it can change to the form of a hand. This change is often set on the basis of control of an application program.

[0153]So, when a remote user (it is shown in drawing 11 like) controls the application program which is running on the host machine, an application program can change the cursor on a host machine, without using a GDI call. Therefore, prehension of GDI shown in drawing 15 and 15A is inefficiency to reproducing the cursor which changed on a remote display.

[0154]In order to cope with this problem, this invention supervises the function to change real cursor (for example, the SetCursor command etc.). This invention reproduces cursor change on a remote computer.

[0155]It is also one method to execute the same SetCursor command. The method instead of another will be changing remote cursor, when host cursor's changes, or making a bit map draw by performing the sequence of a suitable GDI call.

[0156]9. The whole display is not reproduced. This invention only reproduces the window which the user of a display identifies. That is, the user can hold a piece of drawing like Note Pad in secret in a period of a meeting. A GDI call uses the task handle (taskhandle). A GDI call will not be shared if a task handle does not refer to the item to share.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1]It is a figure which illustrates three computers connected with the telephone

wire.

[Drawing 2]An example of an example explained in detail is shown in 3 thru/or drawing 14 of a figure. In this example, the computer program operates by various parties and can attach a postscript.

[Drawing 3]When this invention is in postscript mode and one of host users tries the operation of a computer, it is a figure showing how this invention answers.

[Drawing 4]When this invention is in postscript mode and one of host users tries a postscript to a computer, it is a figure showing how this invention answers.

[Drawing 5]When an application mode has this invention and one of host users tries the operation of a computer, it is a figure showing how this invention answers.

[Drawing 6]When this invention is in local postscript mode and one of host users tries the operation of a computer, it is a figure showing how this invention answers.

[Drawing 7]When this invention is in local postscript mode and one of host users tries a postscript to a computer, it is a figure showing how this invention answers.

[Drawing 8]When this invention is in a view mode, it is a figure showing how it answers in a host user.

[Drawing 9]When this invention is in postscript mode and a remote user tries the operation of a computer, it is a figure showing how this invention answers.

[Drawing 10]When this invention is in postscript mode and a remote user tries a postscript to a computer, it is a figure showing how this invention answers.

[Drawing 11]When this invention is in an application mode and a remote user tries the operation of a computer, it is a figure showing how this invention answers.

[Drawing 12]When this invention is in local postscript mode and a remote user tries the operation of a computer, it is a figure showing how this invention answers.

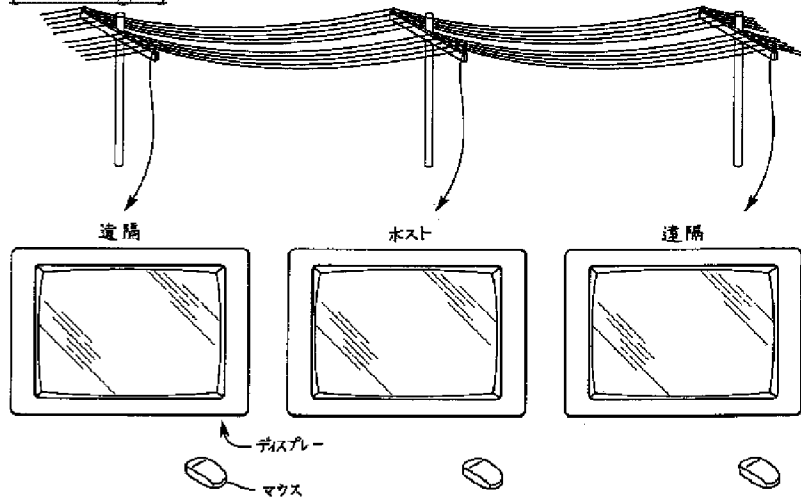
[Drawing 13]When this invention is in local postscript mode and a remote user tries a postscript to a computer, it is a figure showing how this invention answers.

[Drawing 14]When this invention is in a view mode, it is a figure showing how it answers to a remote user.

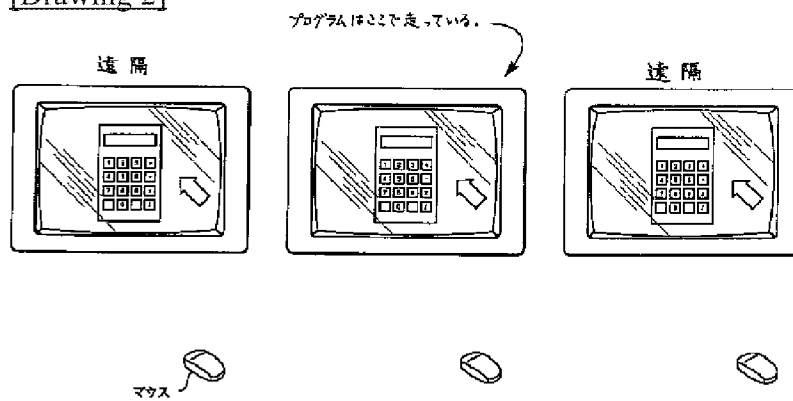
[Drawing 15]It is a figure which illustrates the flow of the logic used by this invention.

DRAWINGS

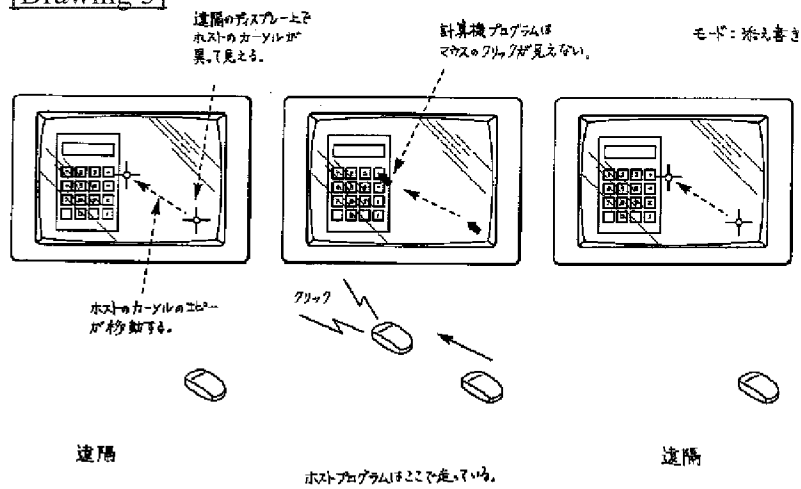
[Drawing 1]



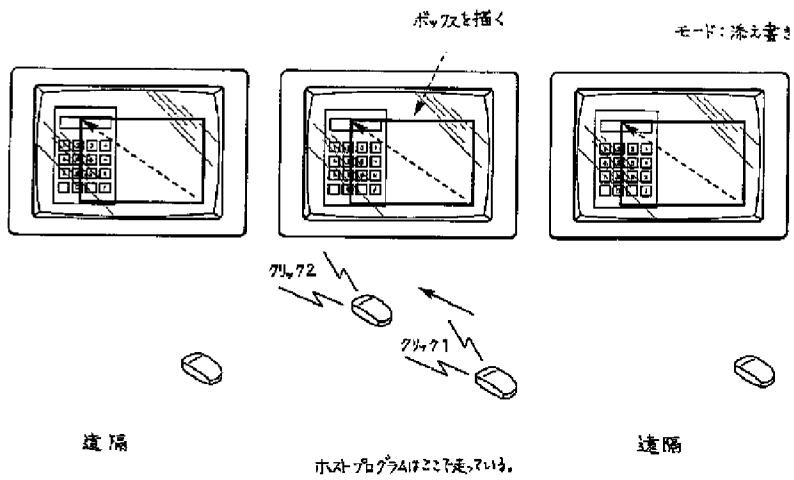
[Drawing 2]



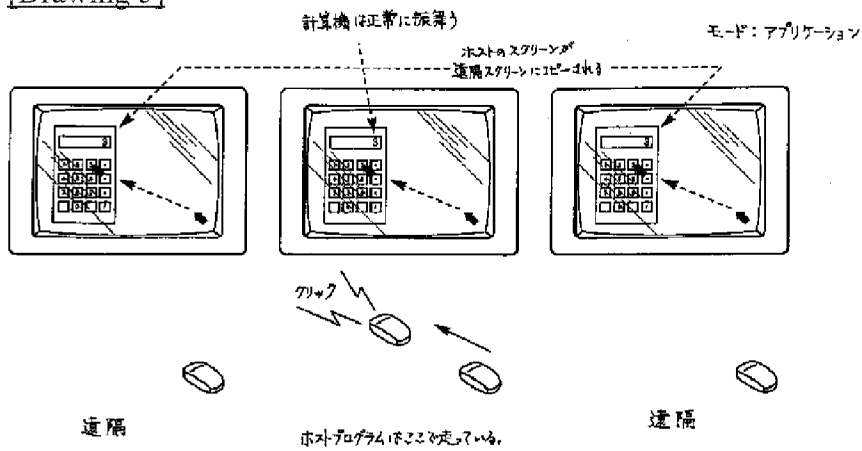
[Drawing 3]



[Drawing 4]

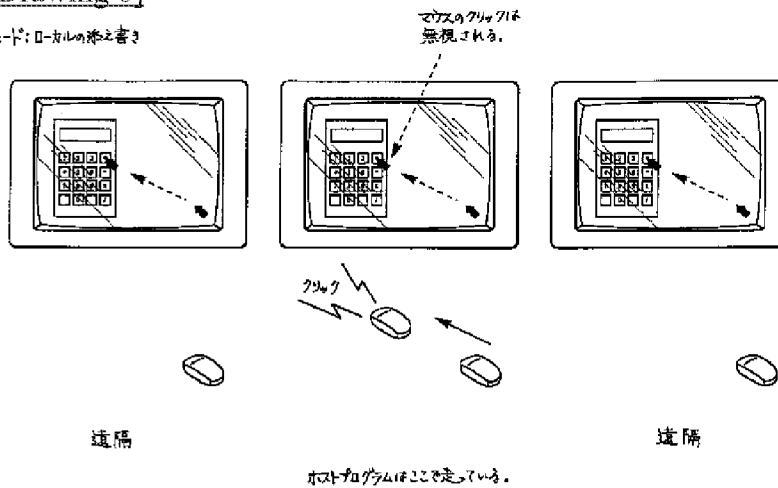


[Drawing 5]

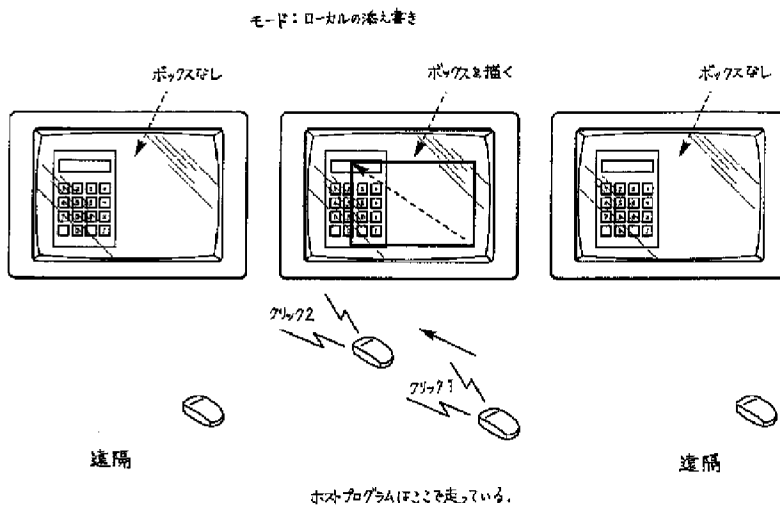


[Drawing 6]

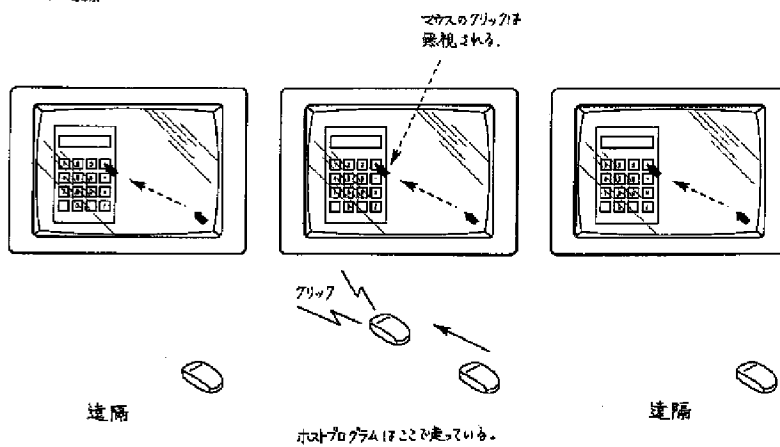
モード: ロールの添え書き



[Drawing 7]



[Drawing 8]
モード: 観望



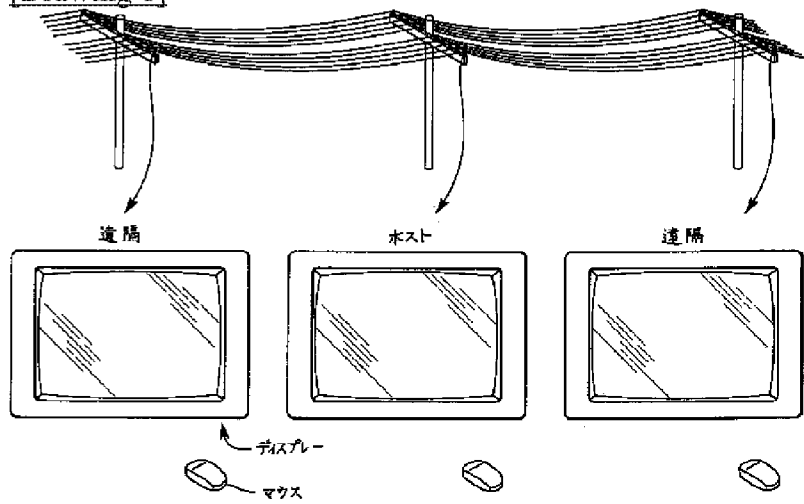
* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

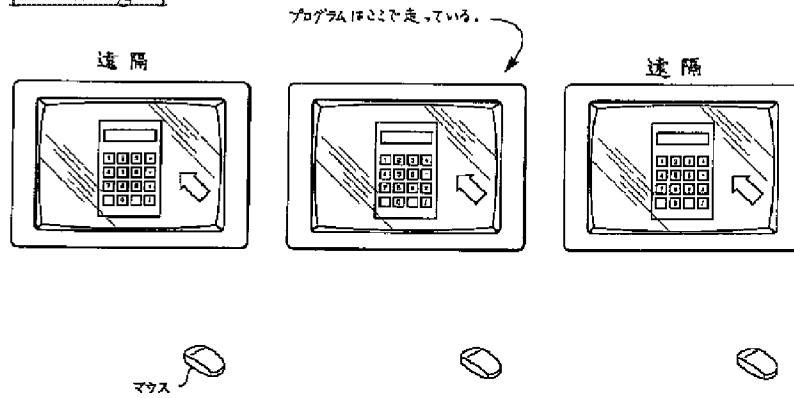
- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DRAWINGS

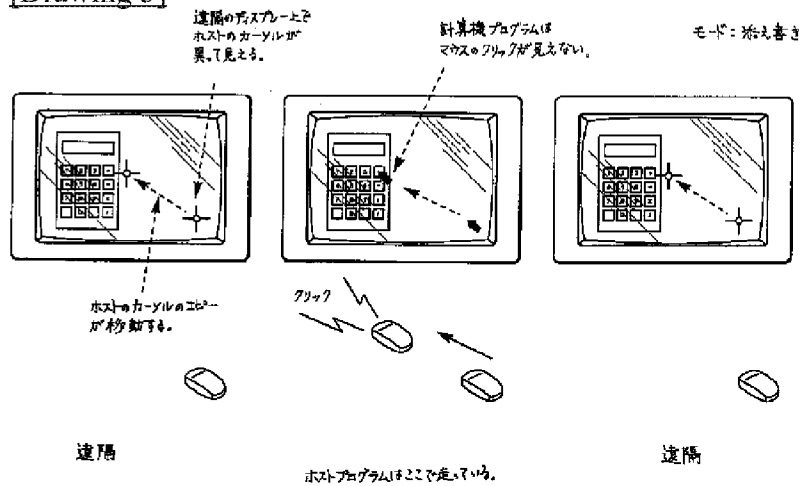
[Drawing 1]



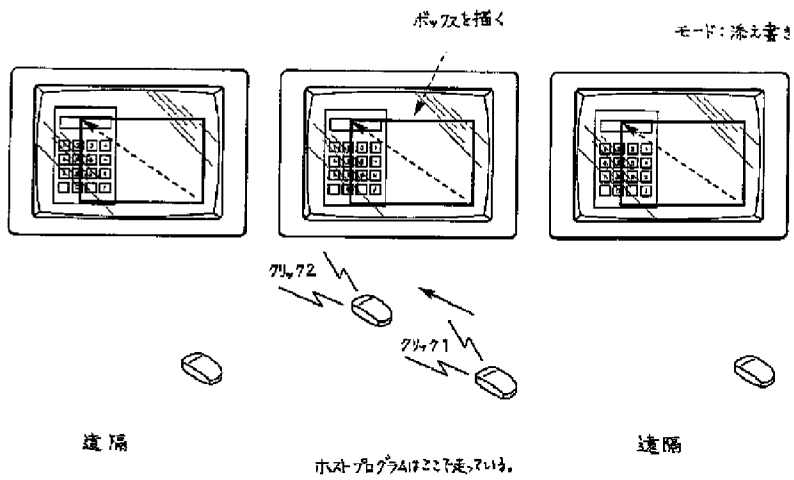
[Drawing 2]



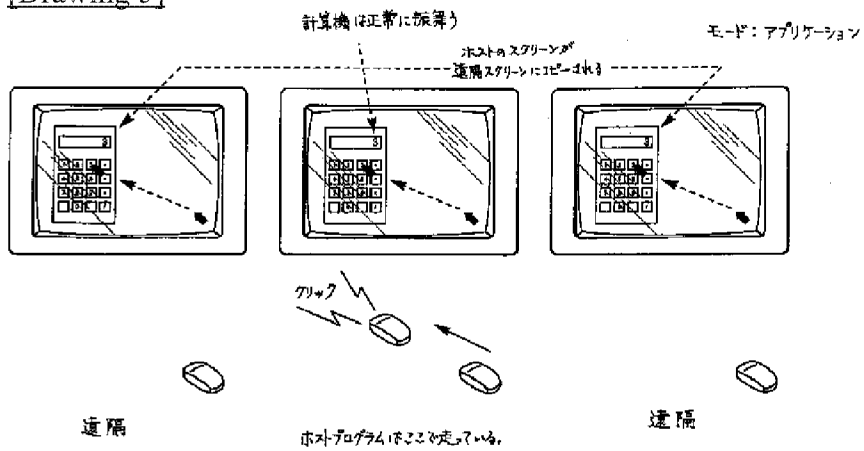
[Drawing 3]



[Drawing 4]

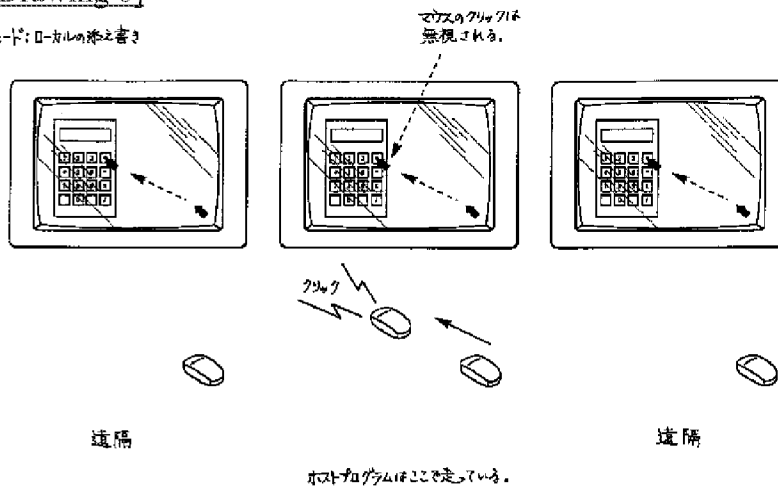


[Drawing 5]



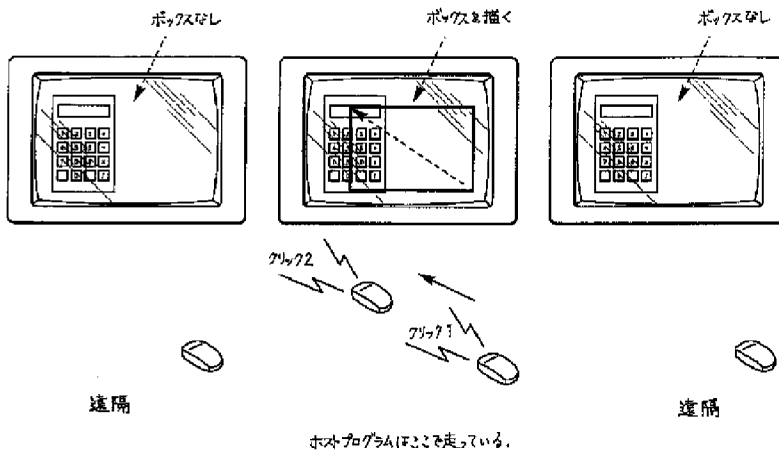
[Drawing 6]

モード: ローカル添え書き



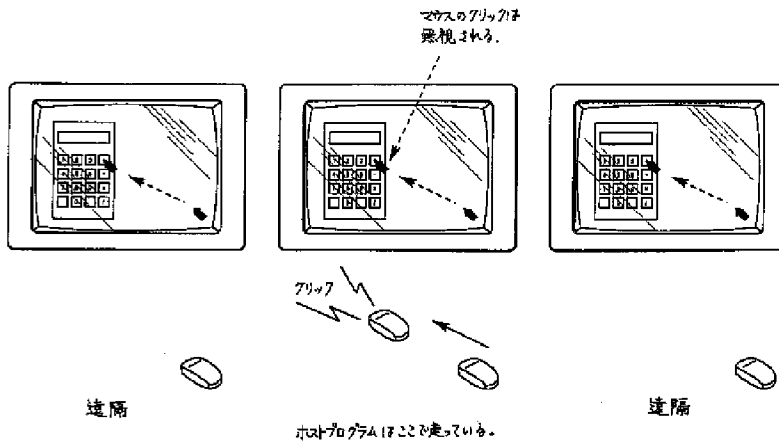
[Drawing 7]

モード：ローカルの書き込み



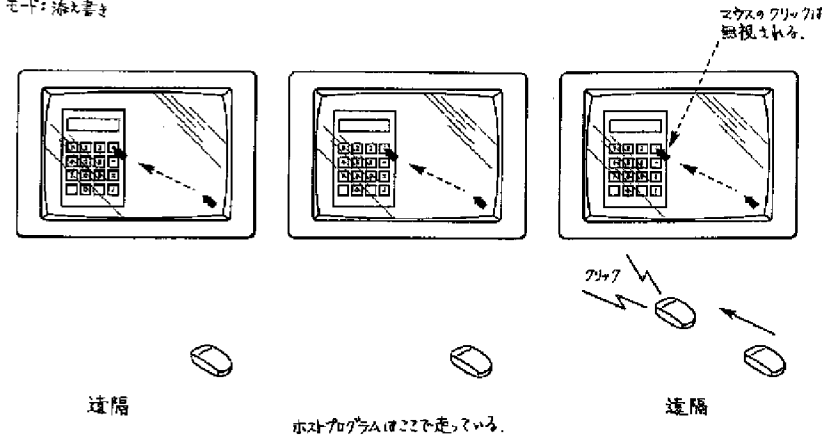
[Drawing 8]

モード：観望



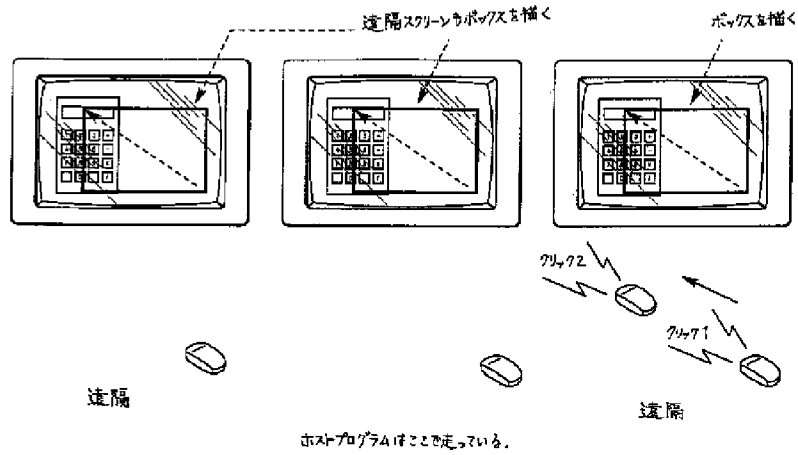
[Drawing 9]

モード：書き込み



[Drawing 10]

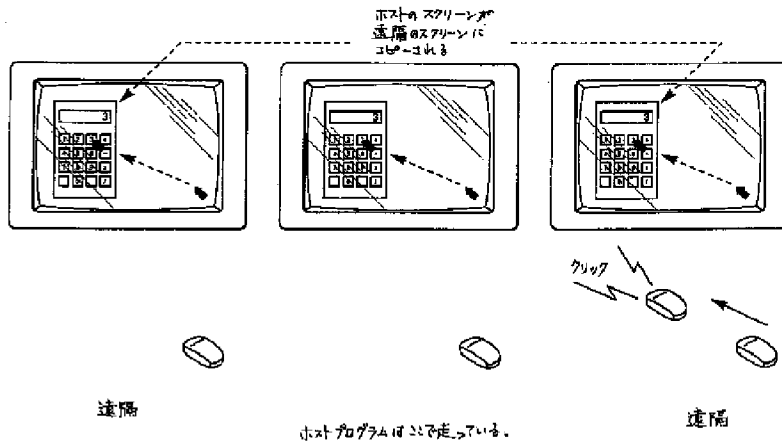
モード: 入力書き



[Drawing 11]

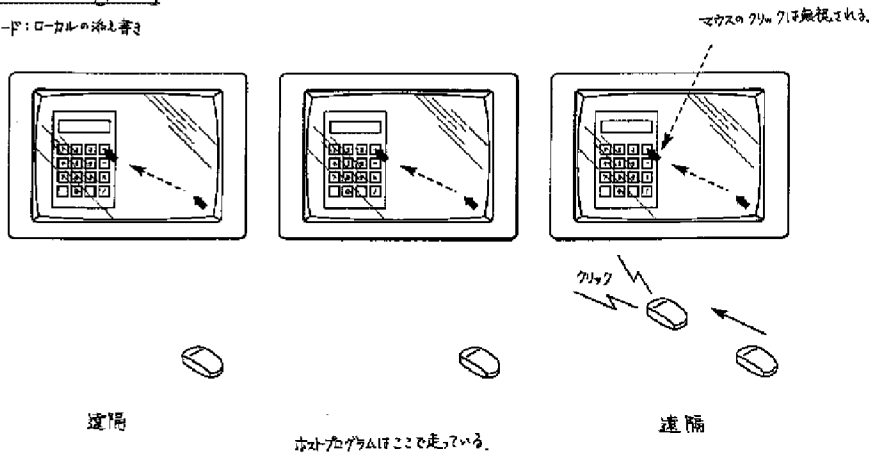
モード: アプリケーション

ホスト上で走っているプログラムは
ホストがボタンとクリックのための
ように振る舞う

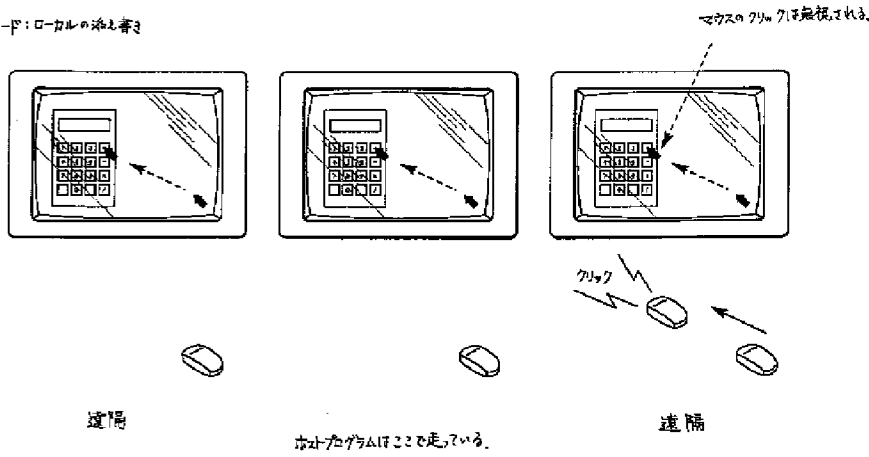


[Drawing 12]

モード：ローカルの読み書き

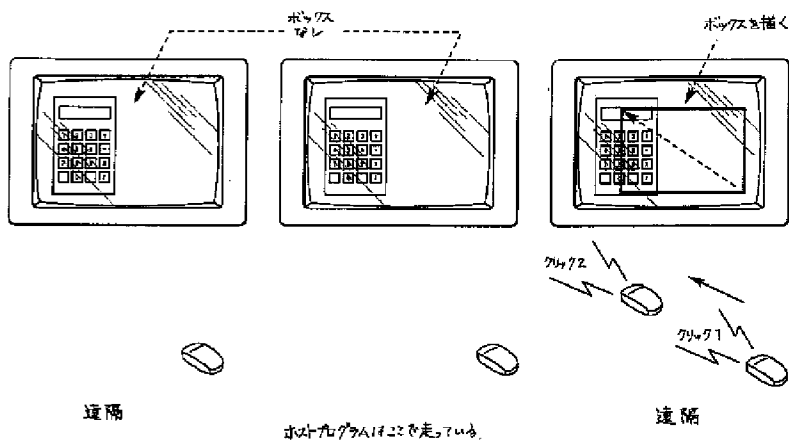


モード：ローカルの読み書き



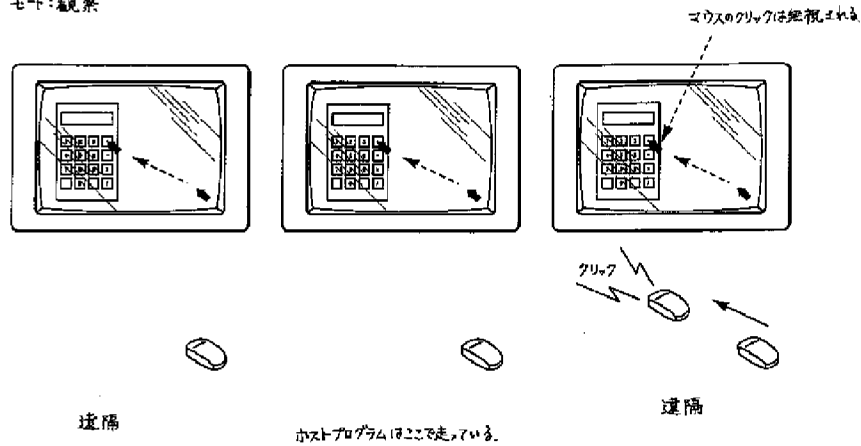
[Drawing 13]

モード：ローカルの読み書き



[Drawing 14]

モード:観察



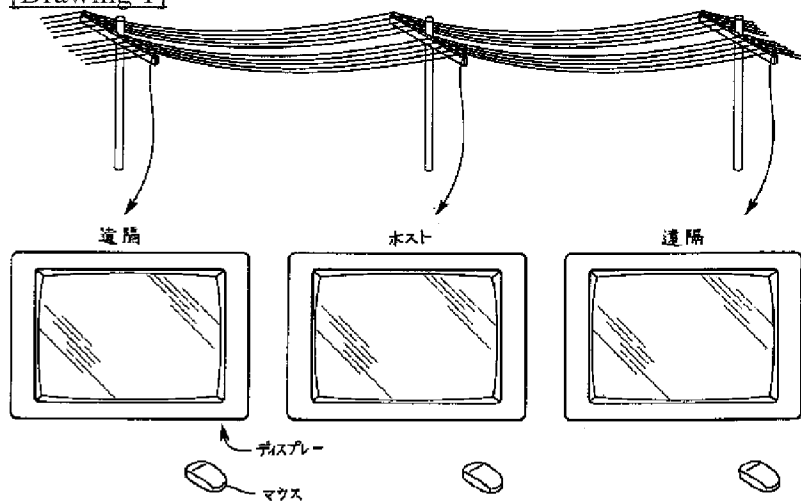
* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

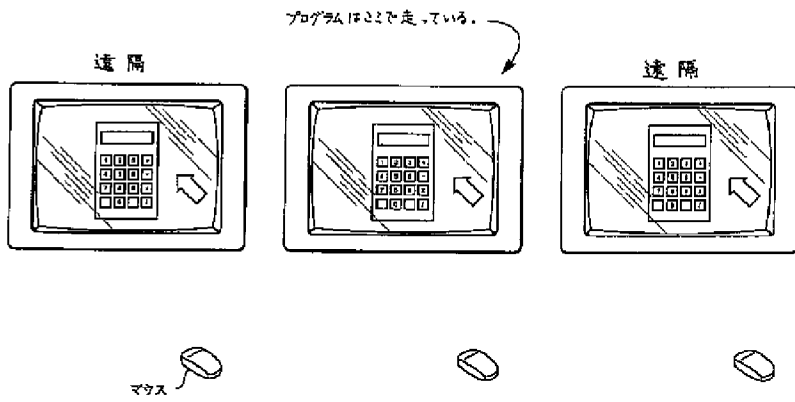
- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DRAWINGS

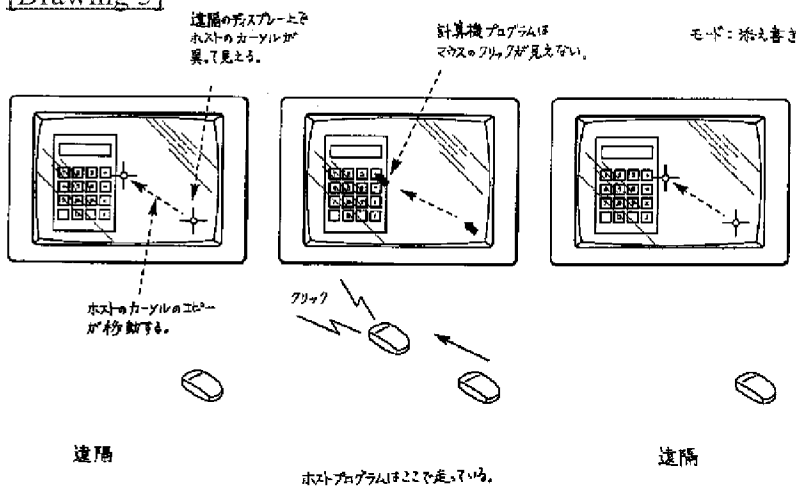
[Drawing 1]



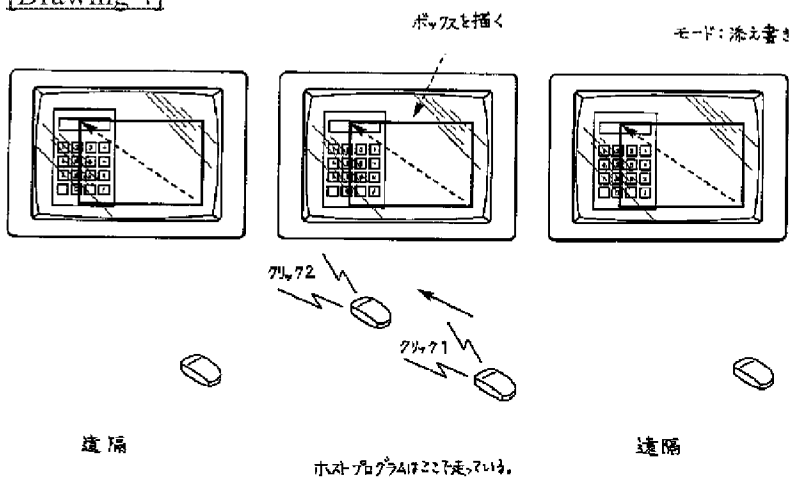
[Drawing 2]



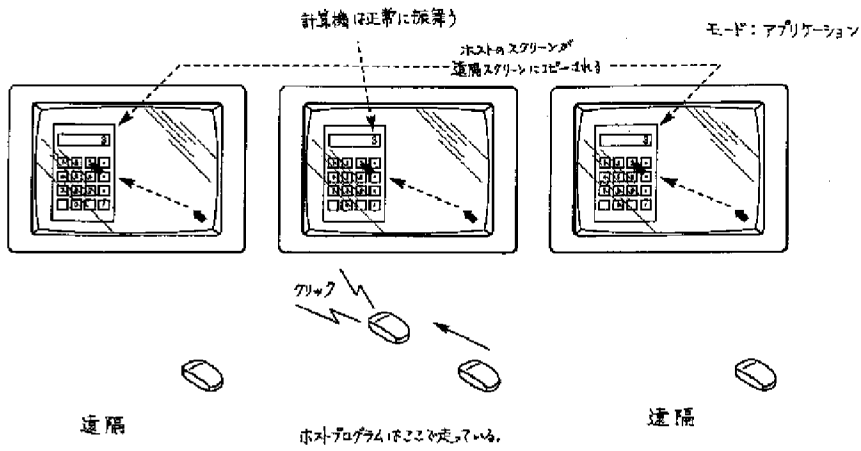
[Drawing 3]



[Drawing 4]

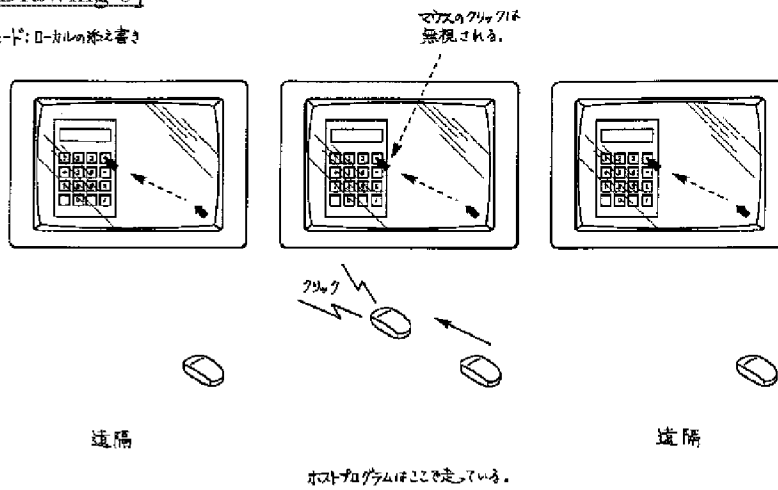


[Drawing 5]



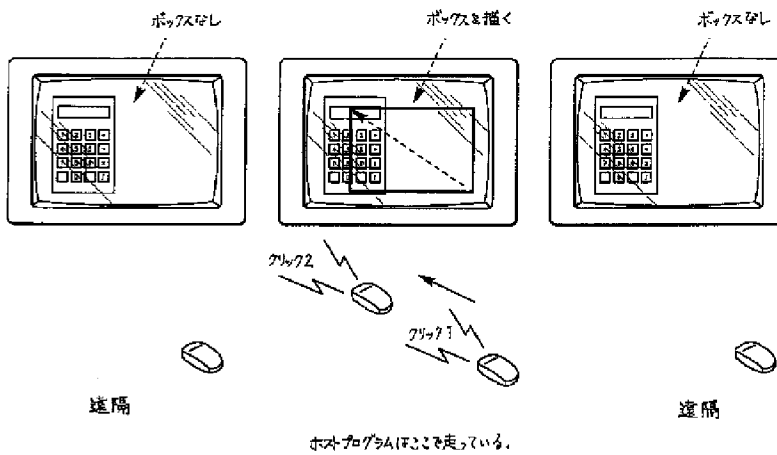
[Drawing 6]

モード：ローカルの添え書き



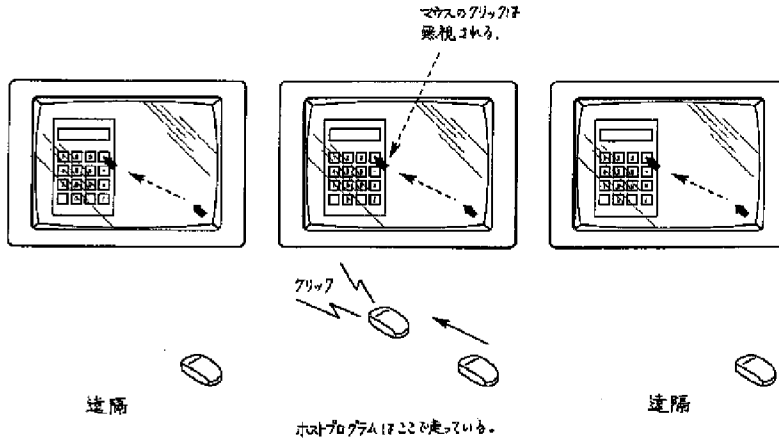
[Drawing 7]

モード：ローカルの添え書き



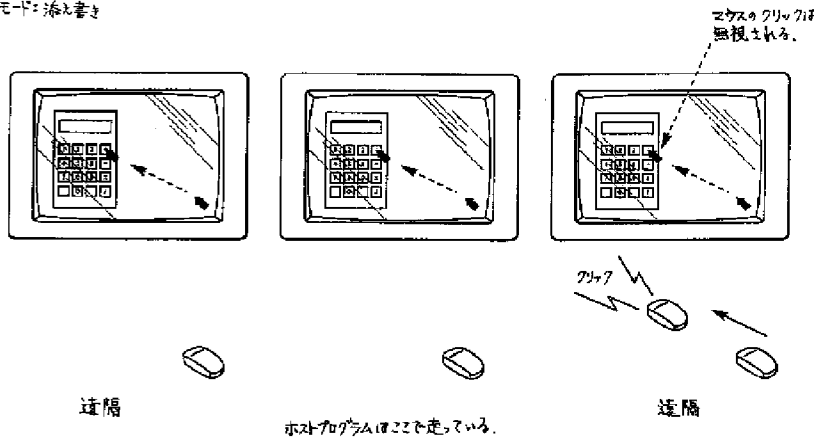
[Drawing 8]

モード: 観察



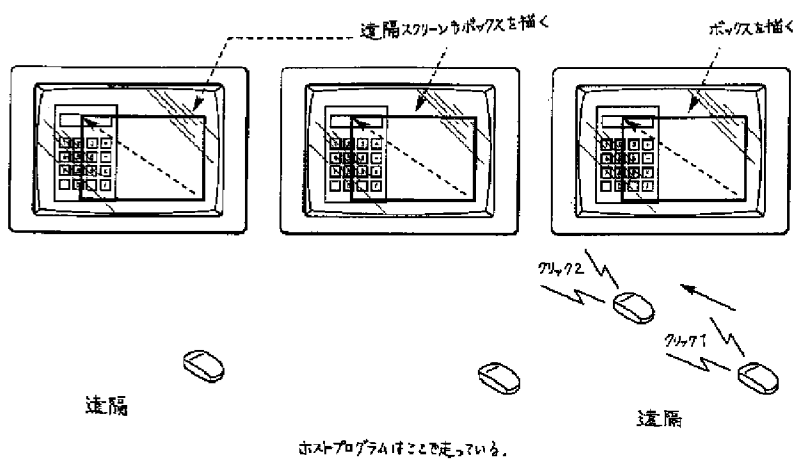
[Drawing 9]

モード: 添え書き



[Drawing 10]

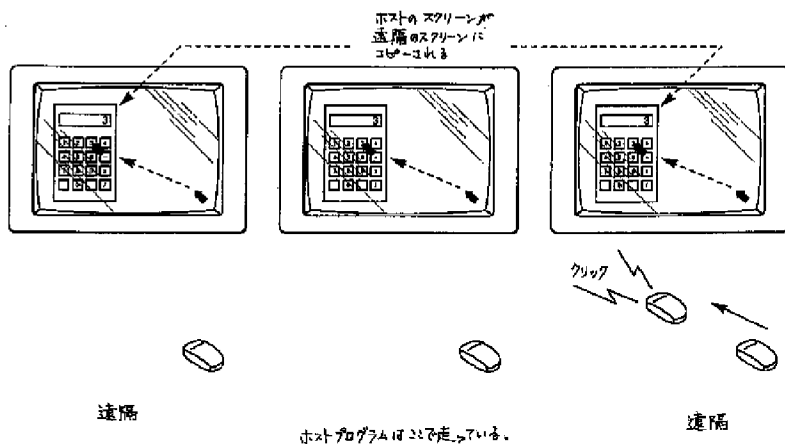
モード: 添え書き



[Drawing 11]

モード: アプリケーション

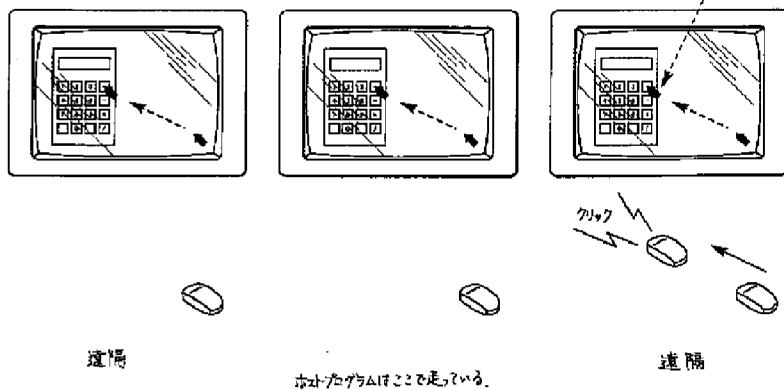
ホスト上で走っているプログラムは
ホストがボタンをクリックするための
ように振舞う



[Drawing 12]

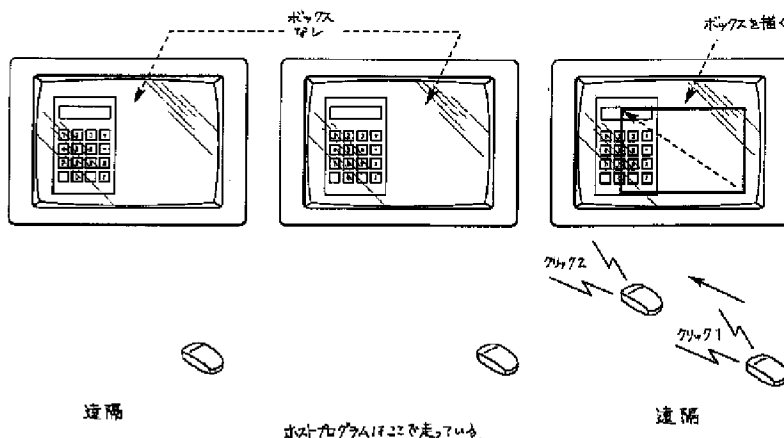
モード: ローカルの読み書き

マウスのクリックは無視される。



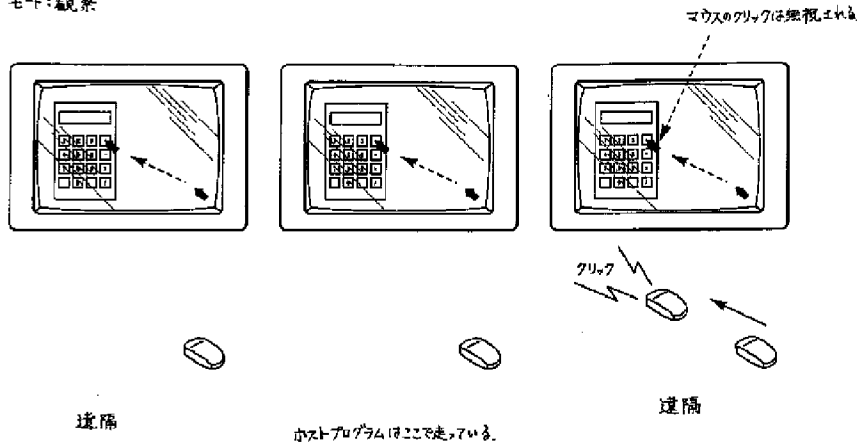
[Drawing 13]

モード: ローカルの読み書き



[Drawing 14]

モード:観察



[Drawing 15]

